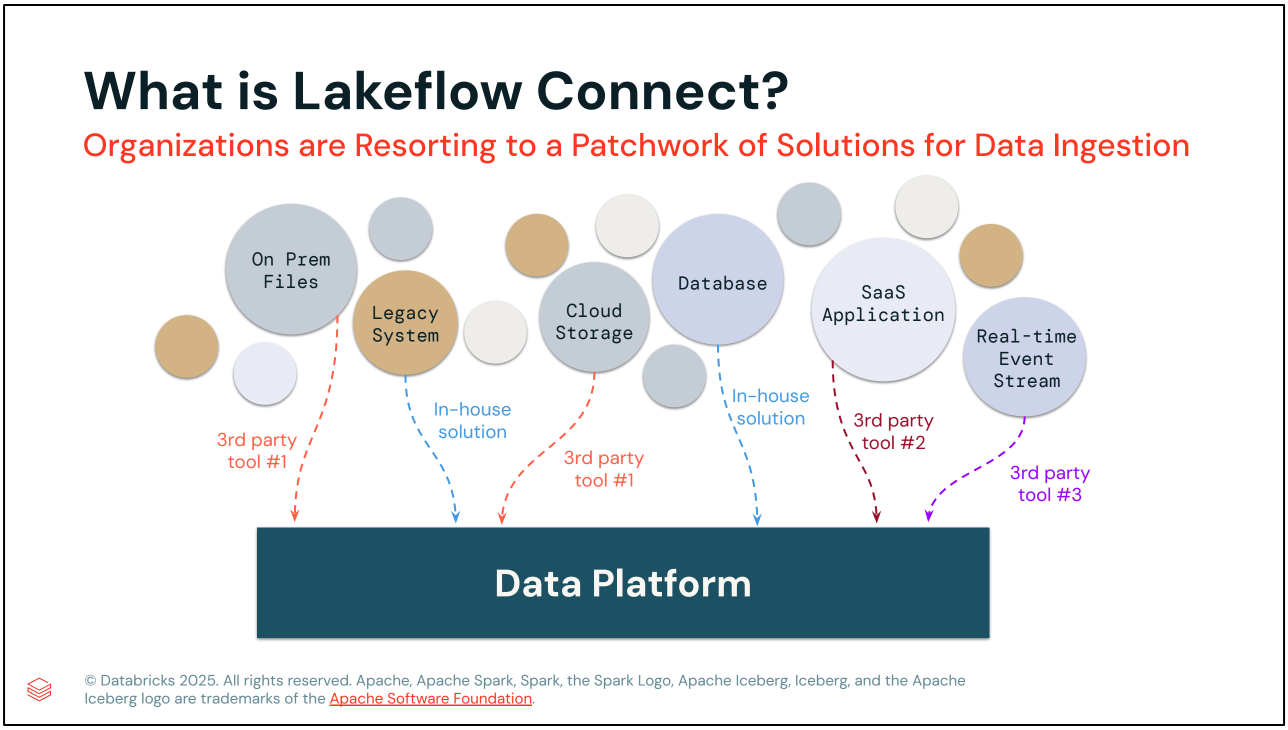
databricks

Introduction to Data Engineering in Databricks

LECTURE

What is Lakeflow Connect?

In Lakeflow Connect, data ingestion is streamlined with simple, efficient connectors that enable you to bring in data from files, cloud storage, databases, enterprise applications, and streaming sources directly into the Databricks Lakehouse—all within a unified, managed platform.



What is Lakeflow Connect?

Organizations are Resorting to a Patchwork of Solutions for Data Ingestion

On Prem Files

Legacy System

Cloud Storage

Database

SaaS Application

Real-time Event Stream

3rd party tool #1

In-house solution

3rd party tool #1

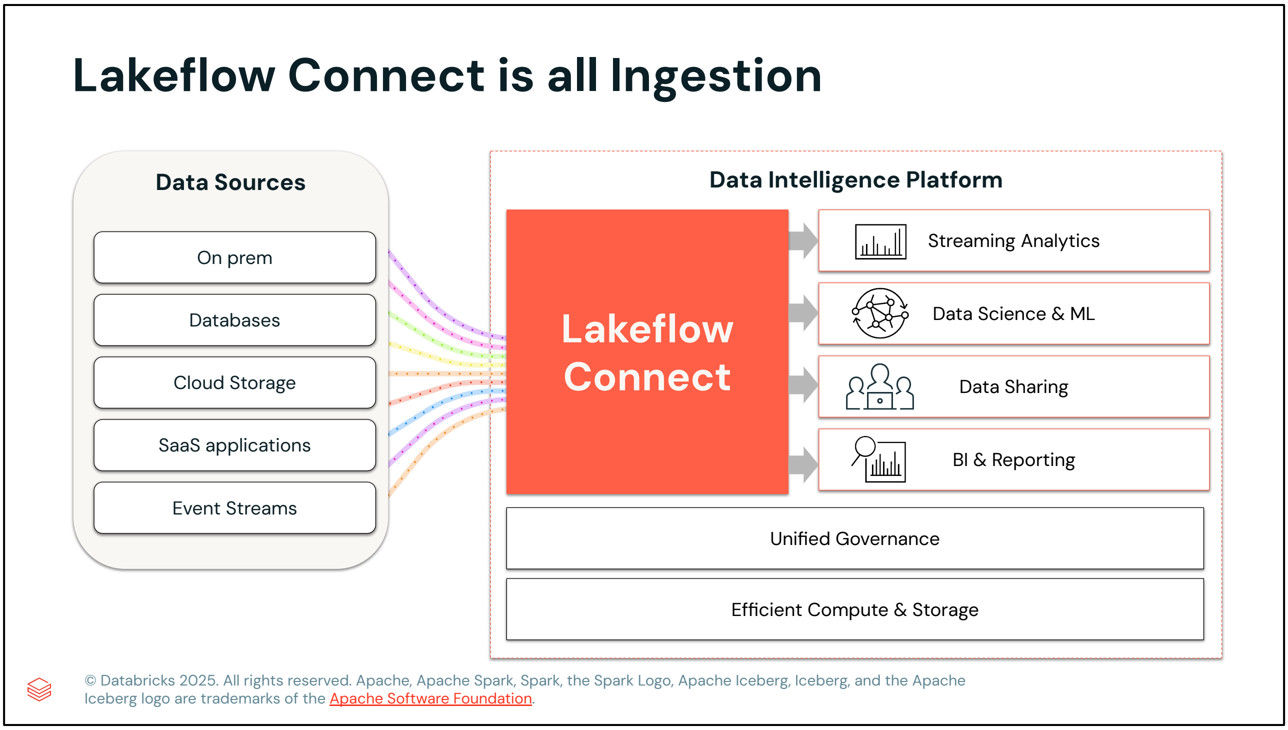
In-house solution

3rd party tool #2

3rd party tool #3

Data Platform

Traditionally, organizations are resorting to a patchwork of solutions for data ingestion when working with enterprise systems, cloud storage and streaming.



Lakeflow Connect is all Ingestion

Data Sources

On prem

Databases

Cloud Storage

SaaS applications

Event Streams

Data Intelligence Platform

Lakeflow Connect

Streaming Analytics

Data Science & ML

Data Sharing

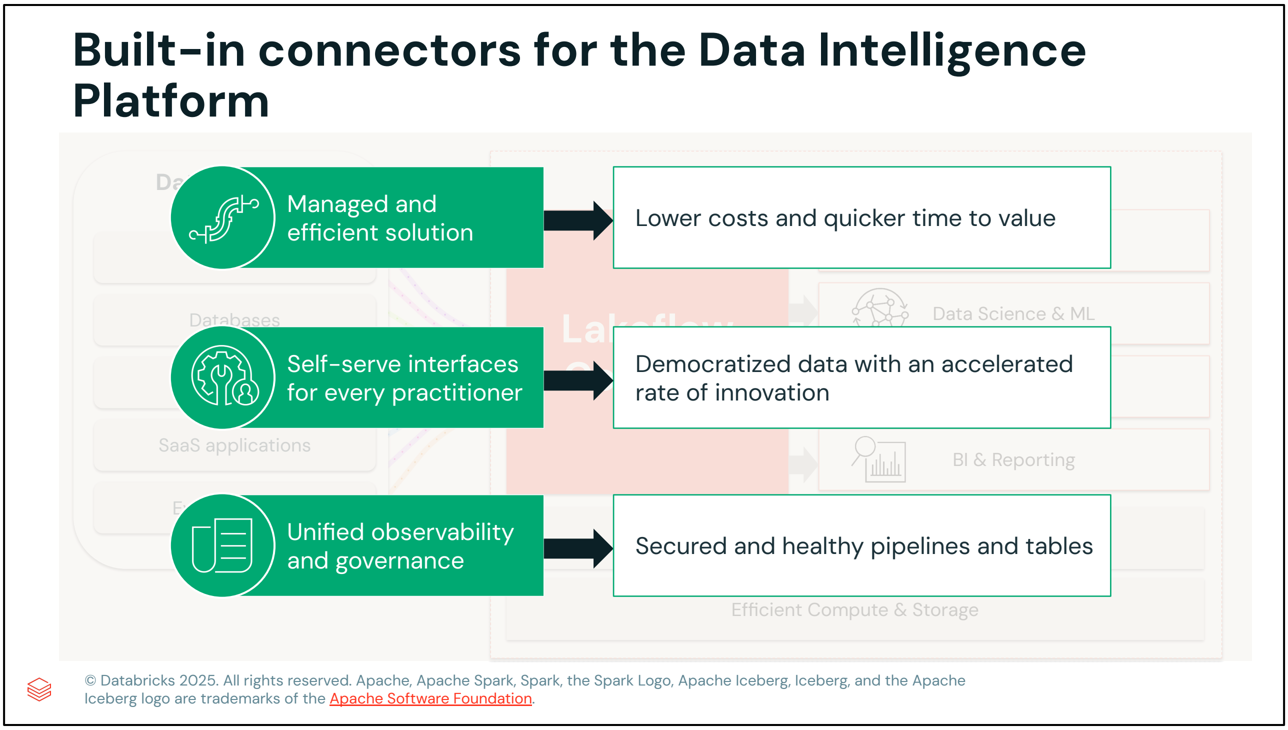
BI & Reporting

Unified Governance

Efficient Compute & Storage

With LakeFlow Connect, you can perform efficient ingestion pipelines all within Databricks.

It's simple setup and maintenance, providing Unified orchestration, observability, and governance all within the Databricks Data Intelligence Platform.



Built-in connectors for the Data Intelligence Platform

Managed and efficient solution → Lower costs and quicker time to value

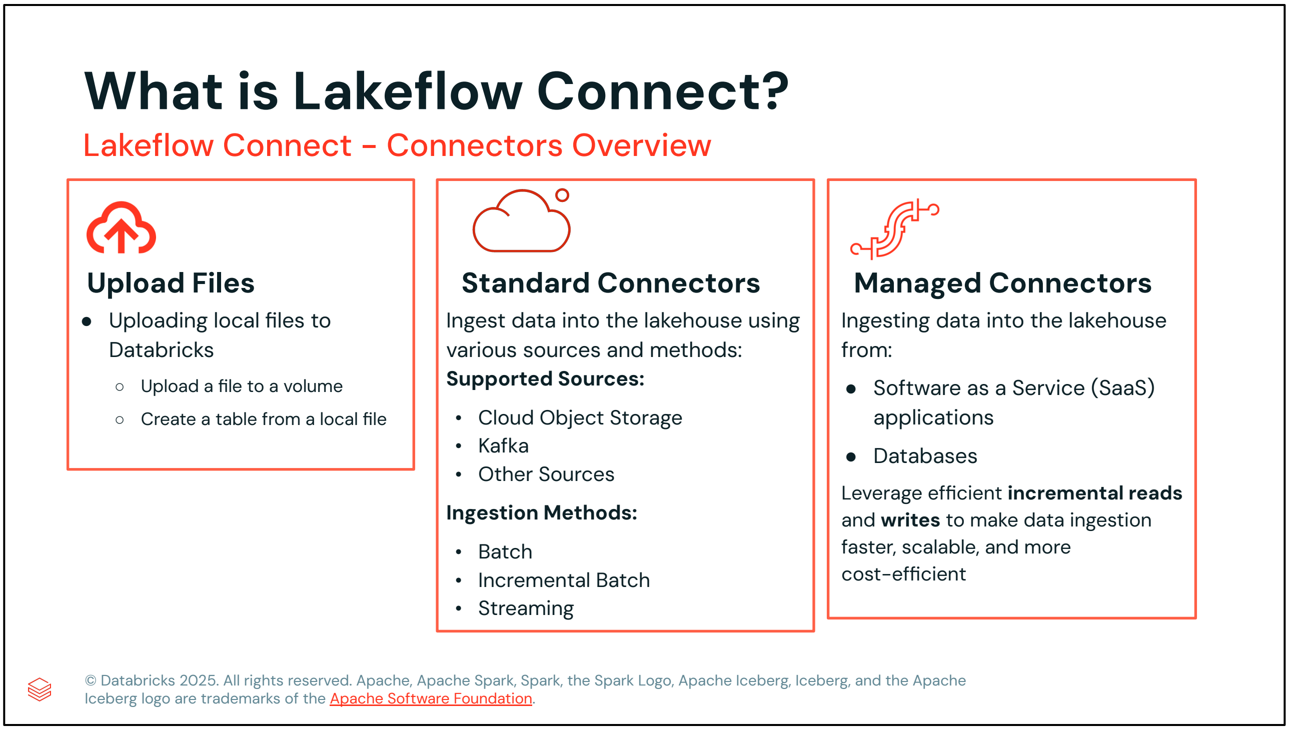
Self-serve interfaces for every practitioner → Democratized data with an accelerated rate of innovation

Unified observability and governance → Secured and healthy pipelines and tables

Lakeflow Connect provides built-in connectors for the Databricks Data Intelligence Platform to streamline data ingestion.

Key benefits include:

- A managed and efficient solution that reduces costs and accelerates time to value.
- Self-service interfaces that enable practitioners across the organization to easily ingest data from enterprise applications.
- Unified observability and governance to ensure secure, reliable, and well-monitored pipelines and tables.



What is Lakeflow Connect?

Lakeflow Connect – Connectors Overview

Upload Files

- Uploading local files to Databricks
  - Upload a file to a volume
  - Create a table from a local file

Standard Connectors

Ingest data into the lakehouse using various sources and methods:

Supported Sources:

- Cloud Object Storage
- Kafka
- Other Sources

Ingestion Methods:

- Batch
- Incremental Batch
- Streaming

Managed Connectors

Ingesting data into the lakehouse from:

- Software as a Service (SaaS) applications
- Databases

Leverage efficient **incremental reads** and **writes** to make data ingestion faster, scalable, and more cost-efficient

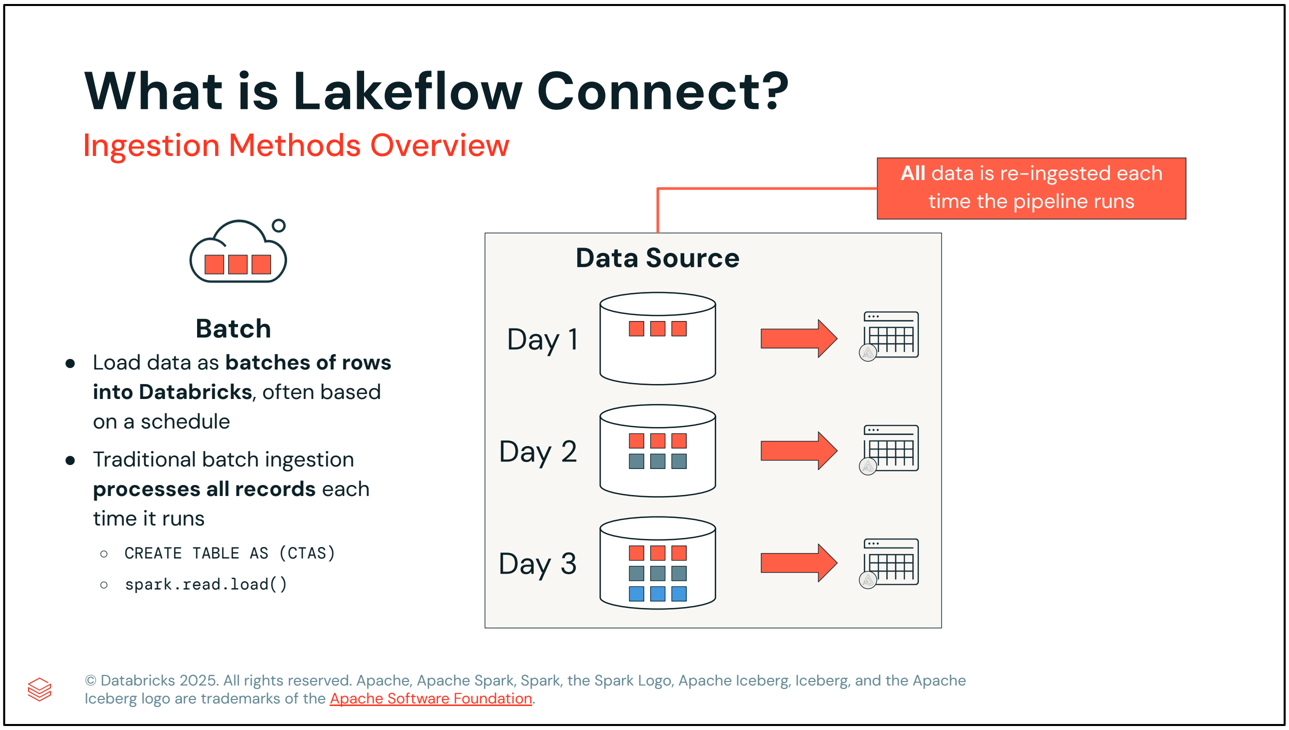
So what exactly is Lakeflow Connect?

Lakeflow Connect provides simple, efficient connectors to ingest data into the Databricks Lakehouse from a wide range of sources, including enterprise applications, databases, cloud storage, local files, message buses, and more. It supports three main types of ingestion:

- **Manual File Uploads:** This allows users to upload local files directly to Databricks into either a volume or as a table, making it extremely easy to bring local data into the platform quickly.

- **Standard Connectors:** These connectors support data ingestion from various sources such as cloud object storage, Kafka, and more. They support multiple ingestion modes, including batch, incremental batch, and streaming. We'll explore these ingestion methods in more detail shortly.

- **Managed Connectors:** Purpose-built for ingesting data from enterprise applications, including SaaS platforms and databases. They leverage efficient incremental read/write patterns to provide scalable, cost-effective, and high-performance data ingestion into the lakehouse.



What is Lakeflow Connect?

Ingestion Methods Overview

Batch

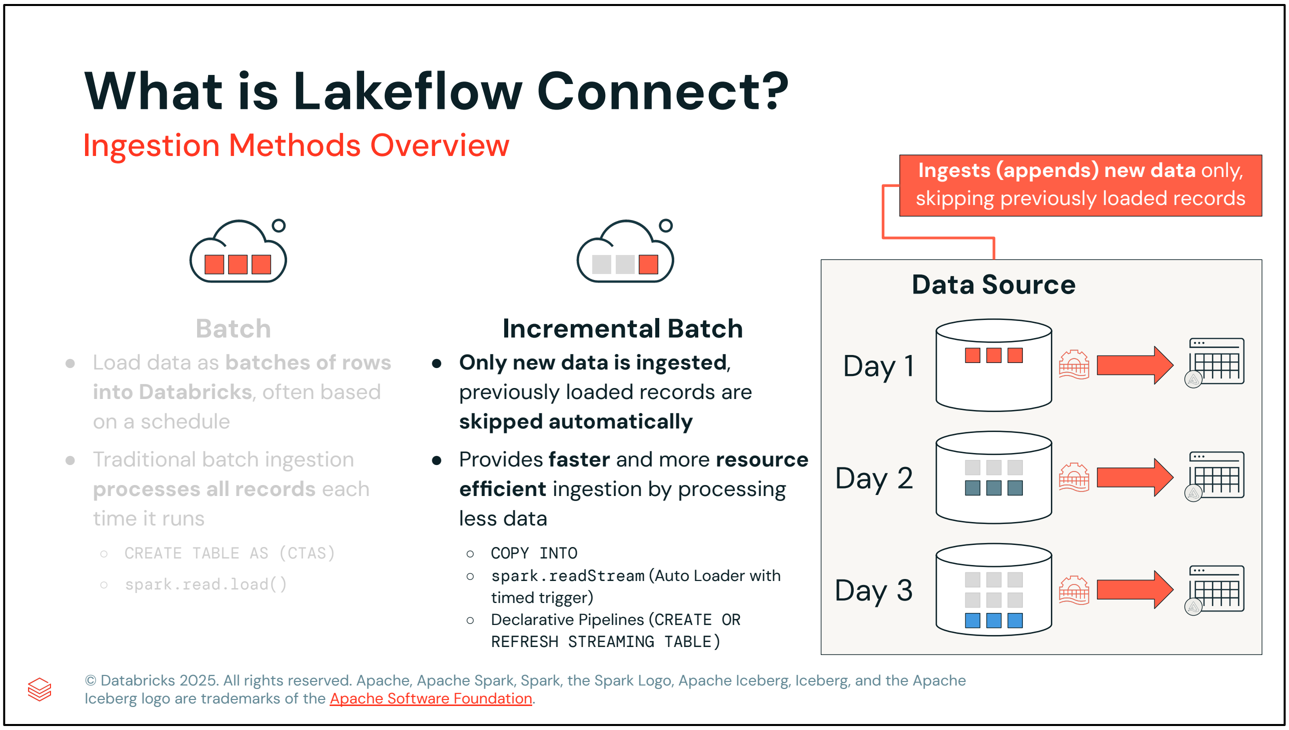
- Load data as **batches of rows into Databricks**, often based on a schedule
- Traditional batch ingestion **processes all records** each time it runs
  - ○ `CREATE TABLE AS (CTAS)`
  - ○ `spark.read.load()`

When ingesting data into Databricks using Lakeflow Connect Standard Connectors, you can choose from several ingestion methods.

Let's start with batch ingestion. Batch ingestion loads data as batches of rows into Databricks, often based on a schedule.

Traditional batch ingestion processes all records each time it runs. Common techniques for performing batch ingestion include:

- **The SQL statement:** CREATE TABLE AS SELECT
- **The Python method:** spark.read.load()



What is Lakeflow Connect?

Ingestion Methods Overview

Batch

- Load data as **batches of rows into Databricks**, often based on a schedule
- Traditional batch ingestion **processes all records** each time it runs
  - `CREATE TABLE AS (CTAS)`
  - `spark.read.load()`

Incremental Batch

- **Only new data is ingested**, previously loaded records are **skipped automatically**
- Provides **faster** and more **resource efficient** ingestion by processing less data
  - `COPY INTO`
  - `spark.readStream` (Auto Loader with timed trigger)
  - Declarative Pipelines (`CREATE OR REFRESH STREAMING TABLE`)

Ingests (appends) new data only, skipping previously loaded records

Data Source

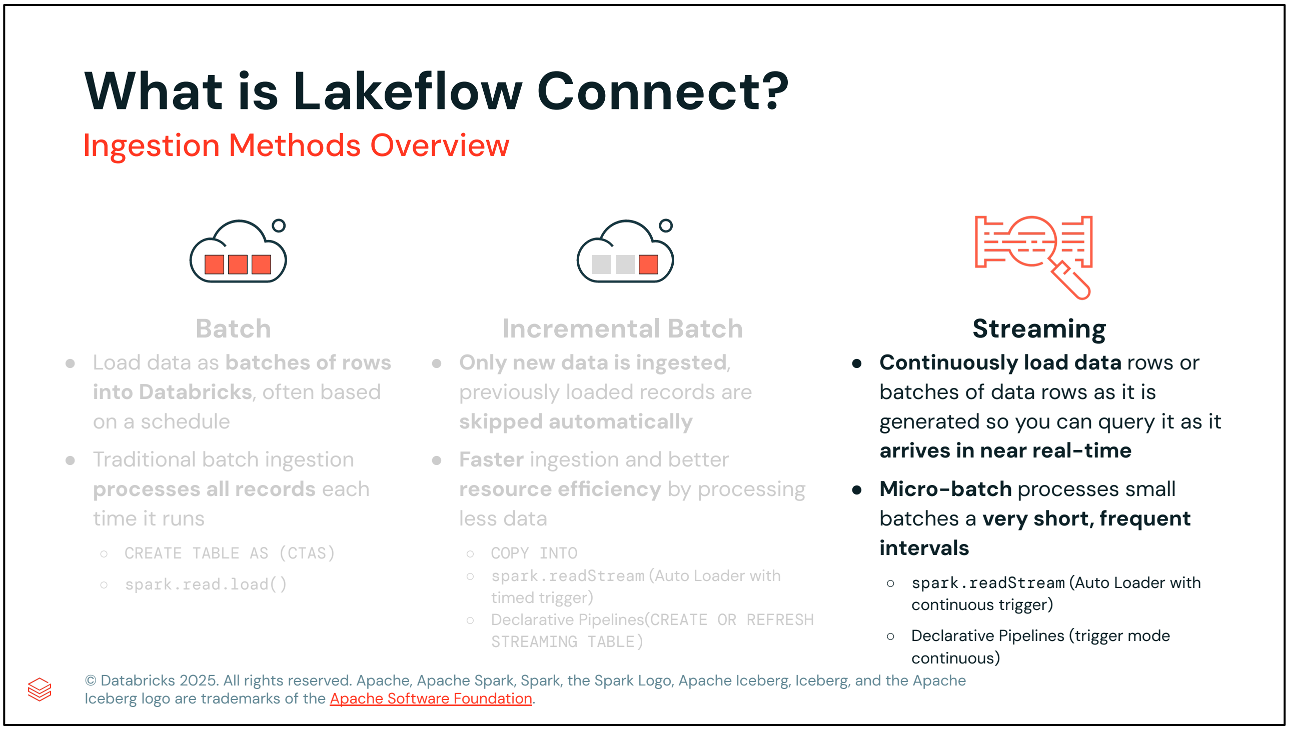
Databricks supports both traditional batch ingestion and incremental batch ingestion options.

While traditional batch ingestion processes all records every time it runs, incremental batch ingestion automatically detects new records in the data source and skips records that have already been ingested. This means only new data is ingested.

Incremental batch ingestion is faster and more resource efficient because it processes only new records instead of reprocessing the entire data source.

Common techniques for performing incremental batch ingestion include:

- The SQL statement: COPY INTO
- The Python method: spark.readStream (Auto Loader with a timed trigger)
- Declarative Pipelines: CREATE OR REFRESH STREAMING TABLE



What is Lakeflow Connect?

Ingestion Methods Overview

Batch

- Load data as **batches of rows into Databricks**, often based on a schedule
- Traditional batch ingestion **processes all records** each time it runs
  - `CREATE TABLE AS (CTAS)`
  - `spark.read.load()`

Incremental Batch

- **Only new data is ingested**, previously loaded records are **skipped automatically**
- **Faster** ingestion and better **resource efficiency** by processing less data
  - `COPY INTO`
  - `spark.readStream` (Auto Loader with timed trigger)
  - Declarative Pipelines(`CREATE OR REFRESH STREAMING TABLE`)

Streaming

- **Continuously load data** rows or batches of data rows as it is generated so you can query it as it **arrives in near real-time**
- **Micro-batch** processes small batches a **very short, frequent intervals**
  - `spark.readStream` (Auto Loader with continuous trigger)
  - Declarative Pipelines (trigger mode continuous)

With streaming ingestion, data is continuously loaded as it is generated, allowing you to query it in near real-time. This method is ideal for loading streaming data from sources such as Apache Kafka, Amazon Kinesis, Google Pub/Sub, and Apache Pulsar.

Streaming ingestion processes data as it arrives, enabling low-latency analysis and immediate action. In contrast, micro-batch ingestion collects data over short, frequent intervals (seconds or minutes) and processes it in small batches. This strikes a balance between latency and system efficiency.

Common techniques for performing streaming ingestion include:

- spark.readStream (Auto Loader with continuous trigger)
- Declarative Pipelines (trigger mode continuous)

databricks

Thank you for completing this lesson and continuing your journey to develop your skills with us.