Apache Iceberg Crash Course

Streaming with Apache Iceberg

Curriculum

July 11: What is a Data Lakehouse and What is a Table Format?

July 16: The Architecture of Apache Iceberg, Apache Hudi and Delta Lake

July 23: The Read and Write Process for Apache Iceberg Tables

Aug 13: Understanding Apache Iceberg’s Partitioning Features

Aug 27: Optimizing Apache Iceberg Tables

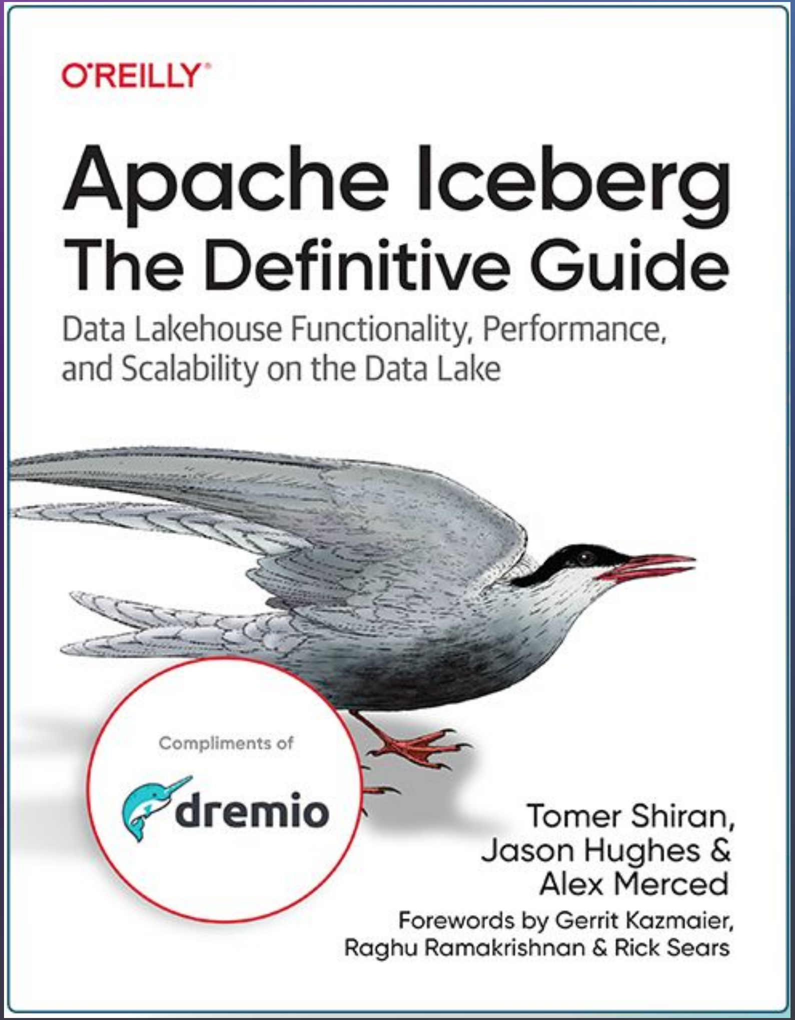
Sep 3: Streaming with Apache Iceberg

Sep 17: The Role of Apache Iceberg Catalogs

Oct 1: Versioning with Apache Iceberg

Oct 15: Ingesting Data into Apache Iceberg with Apache Spark

Oct 29: Ingesting Data into Apache Iceberg with Dremio



O'REILLY

Apache Iceberg
The Definitive Guide

Data Lakehouse Functionality, Performance, and Scalability on the Data Lake

Compliments of dremio

Tomer Shiran, Jason Hughes & Alex Merced

Forewords by Gerrit Kazmaier, Raghu Ramakrishnan & Rick Sears

dremio

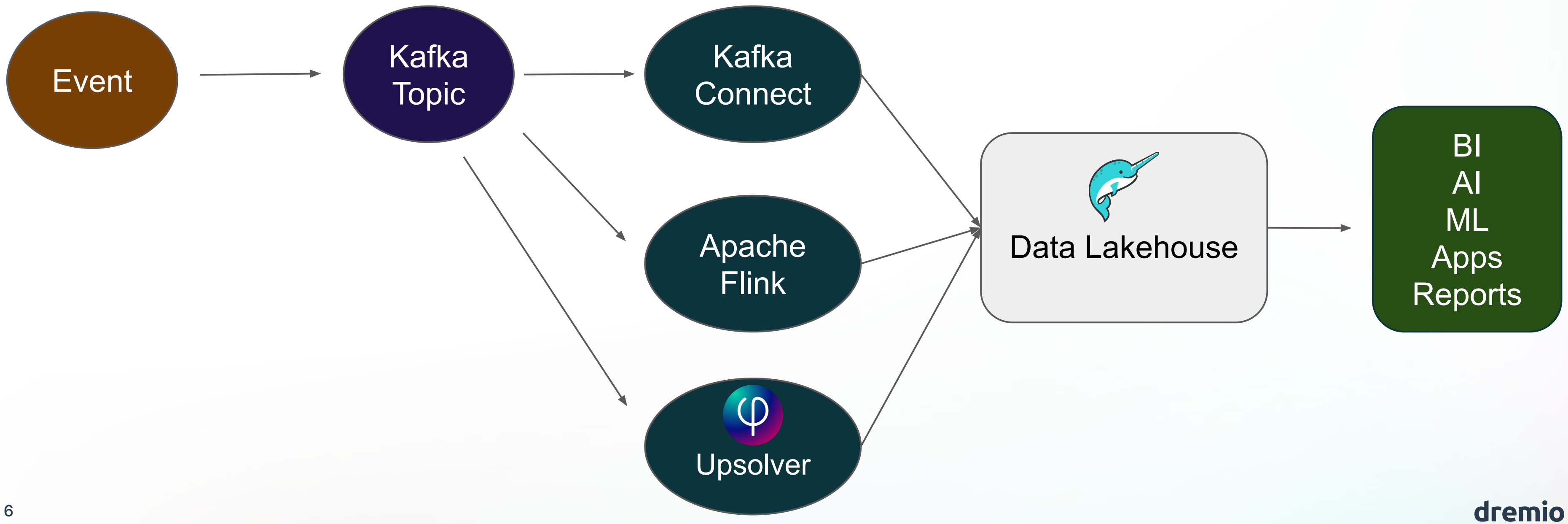
Podcast

dremio.com/gnarly-data-waves
Youtube | Spotify| iTunes

Questions

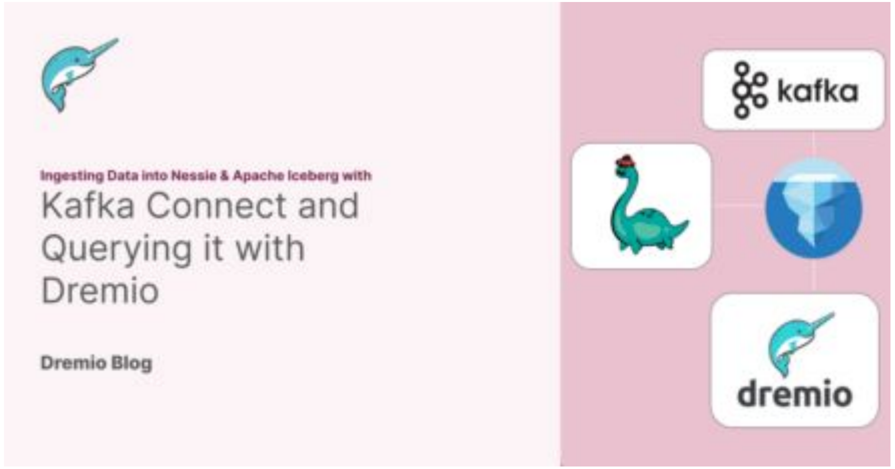
community.dremio.com
Apache Iceberg Category

Streaming to Apache Iceberg



dremio

KAFKA CONNECT


Ingesting Data into Nessie & Apache Iceberg with

Kafka Connect and Querying it with Dremio

Dremio Blog

What is Kafka Connect?

- Kafka Connect is a framework for connecting Kafka with external systems.

- It provides a scalable and reliable way to stream data between Apache Kafka and other data systems.

Key Features:

- **Source Connectors:** Import data from external systems into Kafka topics.

- **Sink Connectors:** Export data from Kafka topics to external systems.

- **Distributed and Standalone Modes:** Run in a distributed mode for scalability and fault tolerance, or standalone for simple, single-node setups.

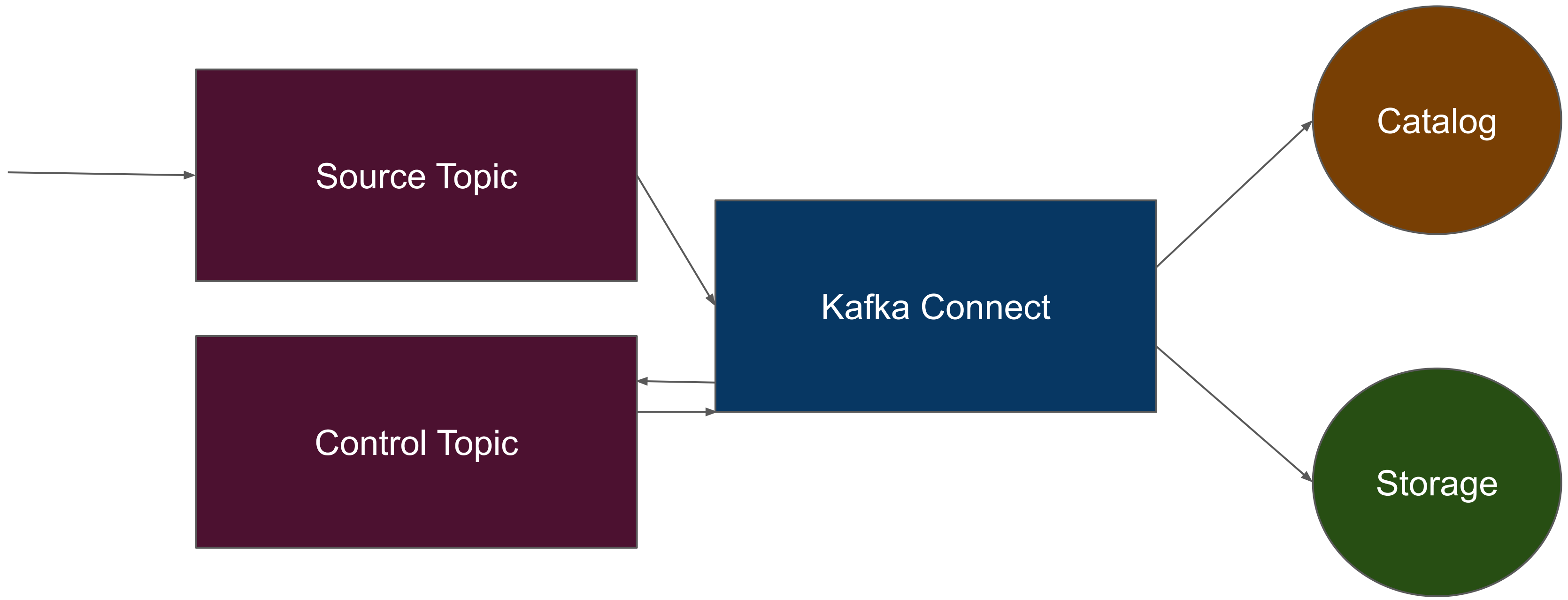
Benefits:

- **Scalability:** Easily scale data pipelines as data volume grows.
- **Flexibility:** Supports a wide variety of source and sink connectors.
- **Simplicity:** Simplifies integration with Kafka, reducing the need for custom coding.

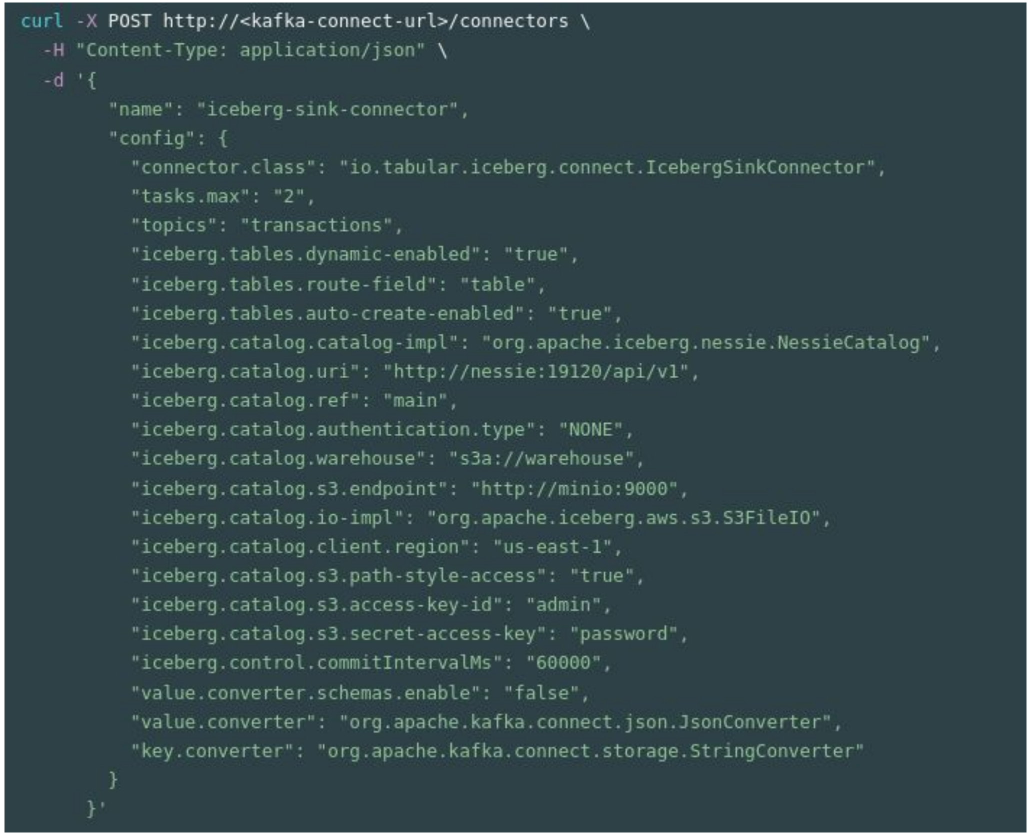
Architecture:

- **Connectors:** Pre-built connectors for various data sources and sinks.
- **Tasks:** Units of work that are configured by connectors.
- **Workers:** Processes that execute tasks and manage data flow.

Structure of Kafka Connect Sink



Configuring a Connector



```
curl -X POST http://<kafka-connect-url>/connectors \
  -H "Content-Type: application/json" \
  -d '{
        "name": "iceberg-sink-connector",
        "config": {
          "connector.class": "io.tabular.iceberg.connect.IcebergSinkConnector",
          "tasks.max": "2",
          "topics": "transactions",
          "iceberg.tables.dynamic-enabled": "true",
          "iceberg.tables.route-field": "table",
          "iceberg.tables.auto-create-enabled": "true",
          "iceberg.catalog.catalog-impl": "org.apache.iceberg.nessie.NessieCatalog",
          "iceberg.catalog.uri": "http://nessie:19120/api/v1",
          "iceberg.catalog.ref": "main",
          "iceberg.catalog.authentication.type": "NONE",
          "iceberg.catalog.warehouse": "s3a://warehouse",
          "iceberg.catalog.s3.endpoint": "http://minio:9000",
          "iceberg.catalog.io-impl": "org.apache.iceberg.aws.s3.S3FileIO",
          "iceberg.catalog.client.region": "us-east-1",
          "iceberg.catalog.s3.path-style-access": "true",
          "iceberg.catalog.s3.access-key-id": "admin",
          "iceberg.catalog.s3.secret-access-key": "password",
          "iceberg.control.commitIntervalMs": "60000",
          "value.converter.schemas.enable": "false",
          "value.converter": "org.apache.kafka.connect.json.JsonConverter",
          "key.converter": "org.apache.kafka.connect.storage.StringConverter"
        }
      }'
```

APACHE FLINK

dremio

Using Flink with Apache Iceberg and Nessie
ICEBERG
DREMIO BLOG

What is Apache Flink?

- Apache Flink is a powerful open-source stream processing framework.

- It enables scalable, high-performance, and fault-tolerant data processing

Key Features:

- **Stream and Batch Processing:** Supports both real-time stream processing and batch processing.

- **Event Time Processing:** Advanced event time processing capabilities for accurate time-based analytics.

- **Stateful Computations:** Efficiently manages state with strong consistency guarantees.

Benefits:

- **High Throughput and Low Latency:** Designed for processing large volumes of data with minimal delay.

- **Fault Tolerance:** Built-in mechanisms for recovering from failures without data loss.

- **Scalability:** Easily scales to handle increasing data loads and complex processing pipelines.

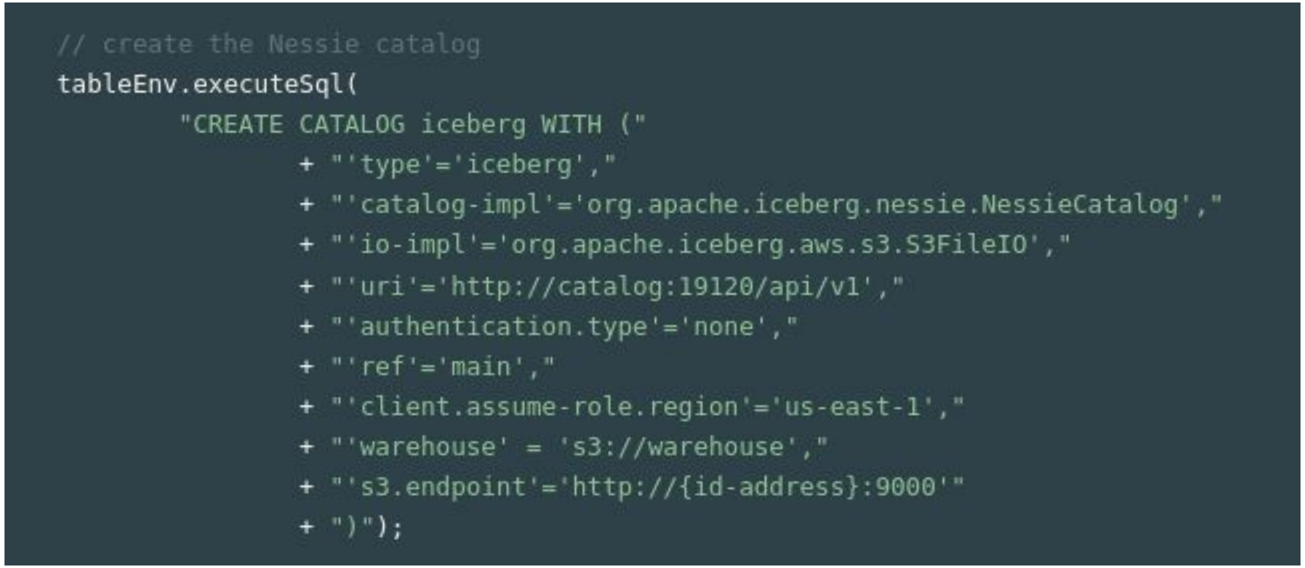
Architecture:

- **Jobs and Tasks:** User-defined jobs broken down into tasks for parallel execution.

- **Operators:** Modular building blocks for defining data transformations.

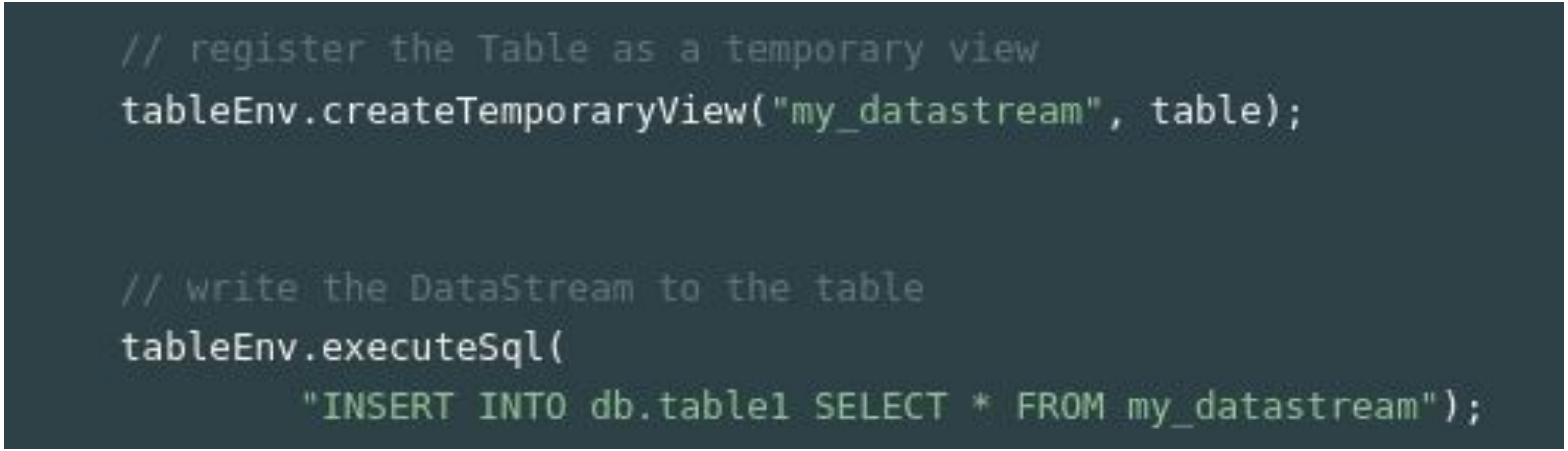
- **State Management:** Efficient state handling using checkpoints and savepoints.

Configuring A Catalog



```
// create the Nessie catalog
tableEnv.executeSql(
        "CREATE CATALOG iceberg WITH ("
                + "'type'='iceberg',"
                + "'catalog-impl'='org.apache.iceberg.nessie.NessieCatalog',"
                + "'io-impl'='org.apache.iceberg.aws.s3.S3FileIO',"
                + "'uri'='http://catalog:19120/api/v1',"
                + "'authentication.type'='none',"
                + "'ref'='main',"
                + "'client.assume-role.region'='us-east-1',"
                + "'warehouse' = 's3://warehouse',"
                + "'s3.endpoint'='http://{id-address}:9000'"
                + ")");
```

Inserting the Data



```
// register the Table as a temporary view
tableEnv.createTemporaryView("my_datastream", table);


// write the DataStream to the table
tableEnv.executeSql(
        "INSERT INTO db.table1 SELECT * FROM my_datastream");
```

UPSOLVER

dremio

Streaming and Batch Data Lakehouses with

Apache Iceberg, Dremio and Upsolver

Dremio Blog

What is Upsolver?

- Upsolver is a cloud-native data lake ETL platform.

- It simplifies the process of ingesting, processing, and preparing data for analytics.

Key Features:

- **No-Code Interface:** User-friendly interface for building data pipelines without writing code.
- **Stream Processing:** Real-time data processing capabilities for continuous data ingestion and transformation.
- **Data Lake Integration:** Native support for data lakes, including AWS S3 and Azure Data Lake Storage.

Benefits:

- **Ease of Use:** Simplifies data engineering tasks with an intuitive, no-code environment.

- **Scalability:** Automatically scales to handle large volumes of data and complex transformations.

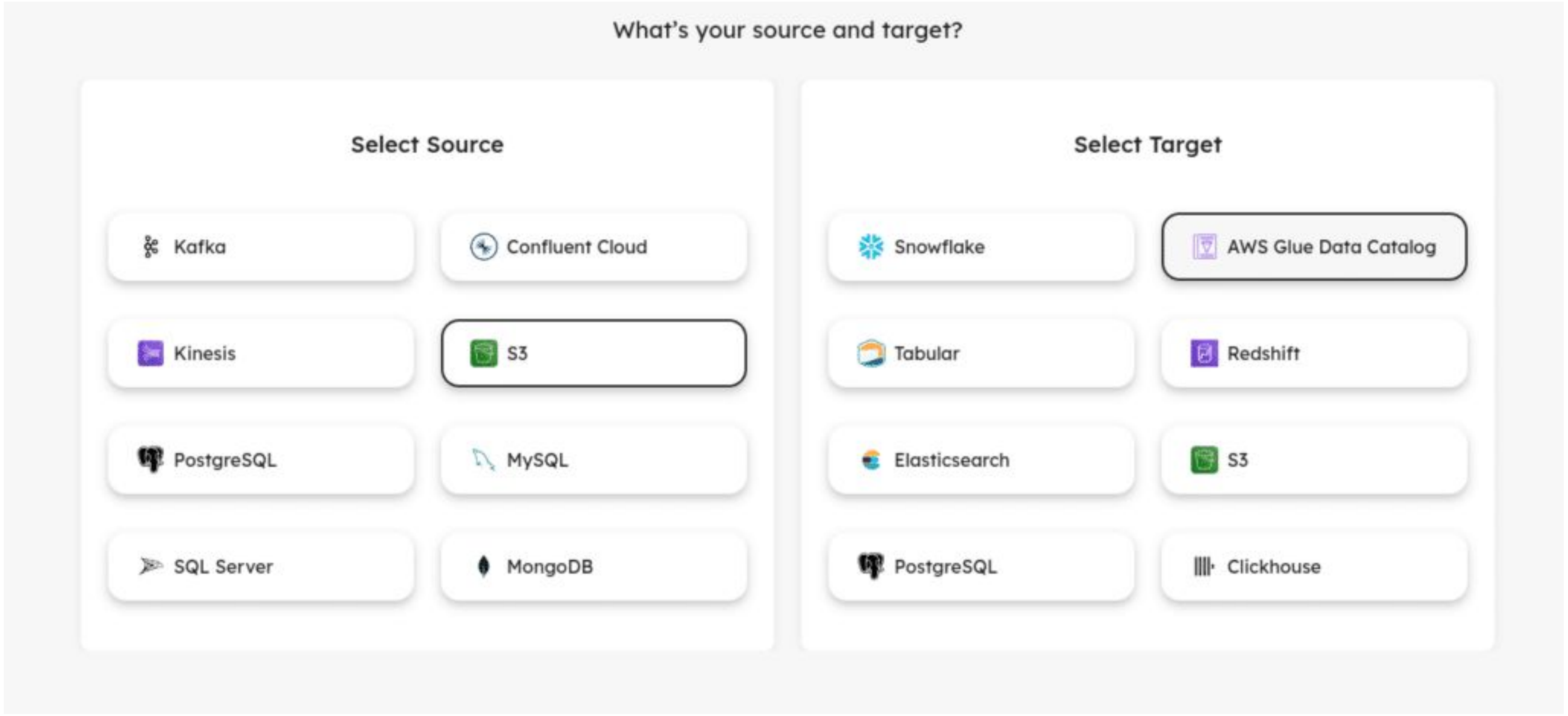
- **Cost Efficiency:** Optimizes storage and compute costs by leveraging cloud infrastructure.

Architecture:

- **Data Ingestion:** Seamlessly ingest data from various sources, including databases, streams, and APIs.

- **Data Transformation:** Apply transformations using SQL-based expressions and functions.

- **Data Output:** Write transformed data to multiple destinations, such as data lakes, warehouses, and analytics platforms.



What's your source and target?

Select Source

Kafka	Confluent Cloud
Kinesis	[✓ S3]
PostgreSQL	MySQL
SQL Server	MongoDB

Select Target

Snowflake	[✓ AWS Glue Data Catalog]
Tabular	Redshift
Elasticsearch	S3
PostgreSQL	Clickhouse

A Iceberg/Dremio Lakehouse on your laptop exercise

Deploy Dremio Software or Dremio Cloud

Hands-On

Postgres -> Iceberg -> Dashboard

SQLServer -> Iceberg -> Dashboard

MongoDB -> Iceberg -> Dashboard

dremio.com/blog

Questions

community.dremio.com
Apache Iceberg Category

dremio