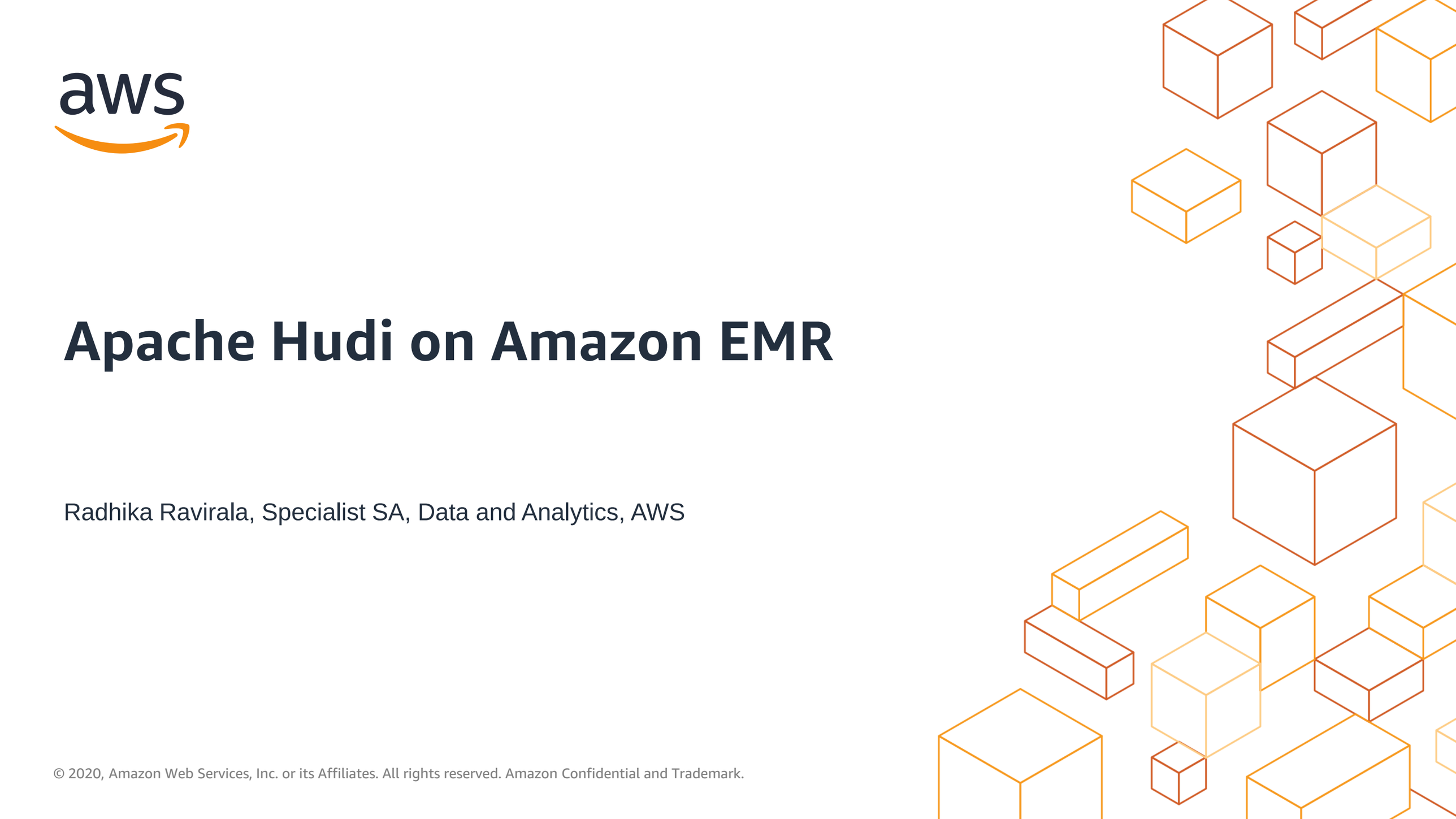


Apache Hudi on Amazon EMR

Radhika Ravirala, Specialist SA, Data and Analytics, AWS

Key takeaways

Motivating use cases

How Apache Hudi works

Architecture patterns

Demo

Next steps

Motivating use cases

Some examples

Change data capture (CDC)

GDPR (data erasure)

De-duplication

Enforce minimum file size on HDFS

Time travel

Change Data Capture & Apply

High value data sits in databases

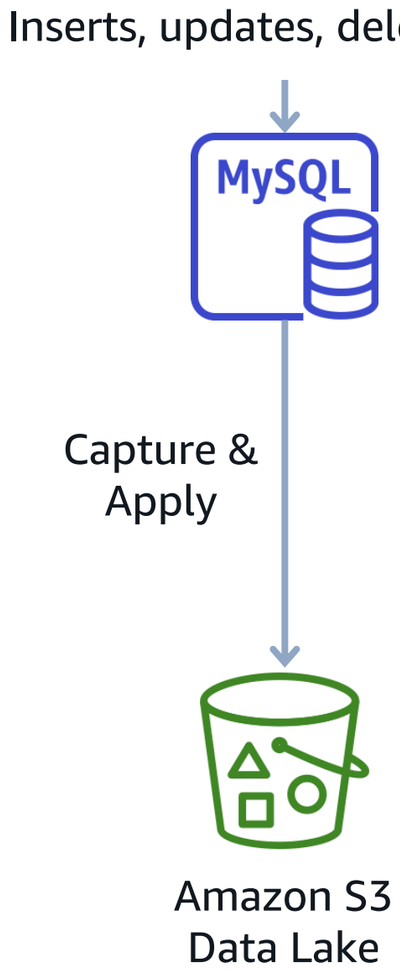
- Most offer CDC and change streams

Applying change streams to Amazon S3 data

- Bulk loads don’t scale
- No support for upserts

Transactionality

- Data quality is a serious concern
- Need similar guarantees as a database



Users

Column	Type
userID	int
country	string
last_modified	long
…	…

Data Deletions: Privacy Regulation

Enforcing data privacy

- Delete data within a specific time frame
- Delete data across all data sets

Find single sentence in a book

- Identifying files & partitions with specific data
- Lack of indexes on data lake storage

Republish the entire book

- Rewriting data is expensive
- Propagating deletes to other systems

Streaming Data Ingestion

Event streams are everywhere

- High volume, time-ordered data
- Duplicate events mess up analytics

Need fast ingestion to Amazon S3

- Schema management, checkpointing
- Write vs. Read optimized storage formats

Balanced approach is important

- Manage file sizes to make queries fast
- Preserve arrival time ordering on S3 storage

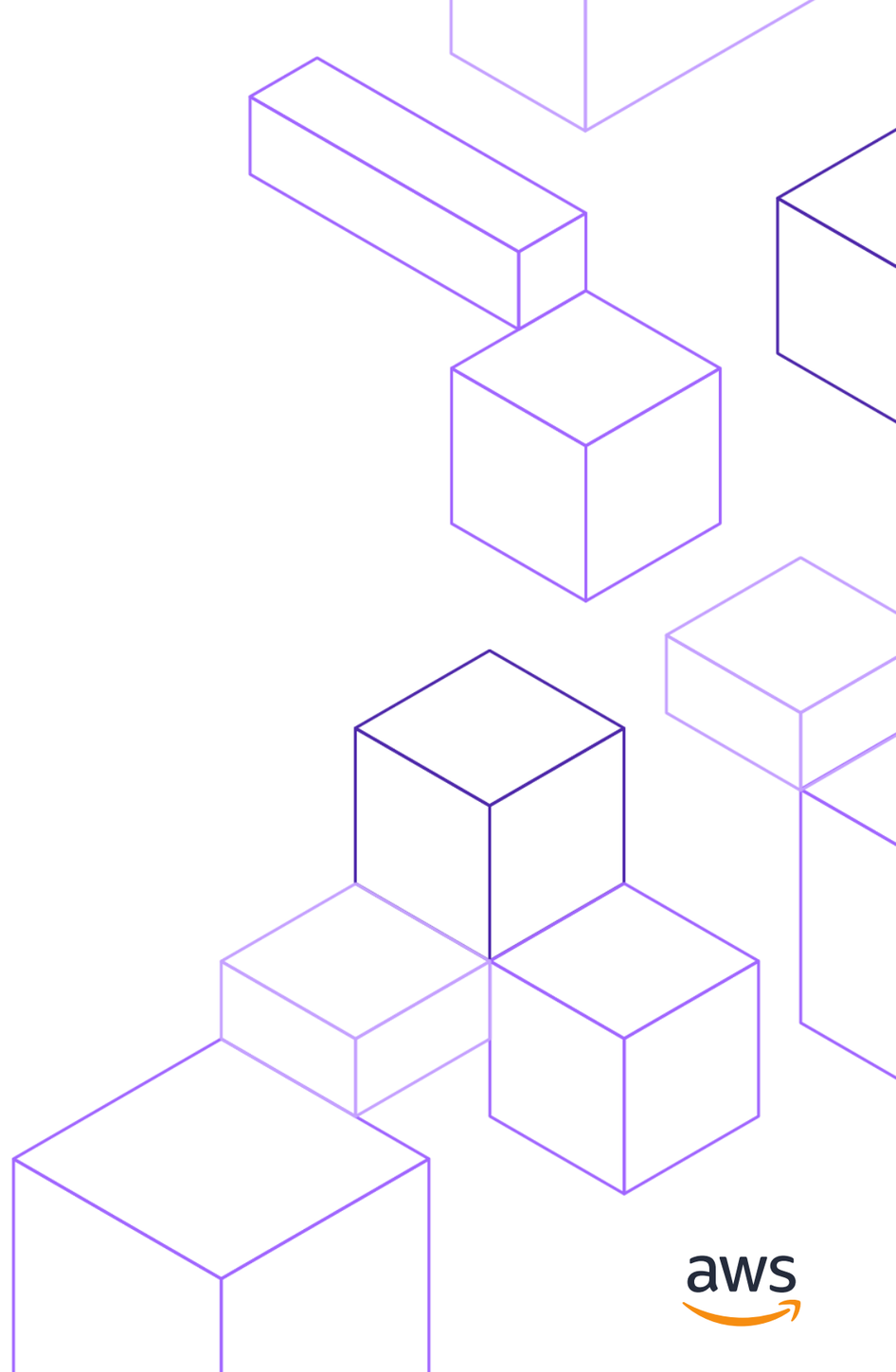
Produce events in Apache Kafka

Amazon S3

Impressions

Field Name	Type
event_id	string
datestr	string
time	long

Apache Hudi



History

2015 : Published core ideas/principles for incremental processing (O'Reilly article)

2016 : Project created at Uber, powers all critical tables @ Uber

2017 : Project open sourced by Uber, powers 100PB data lake

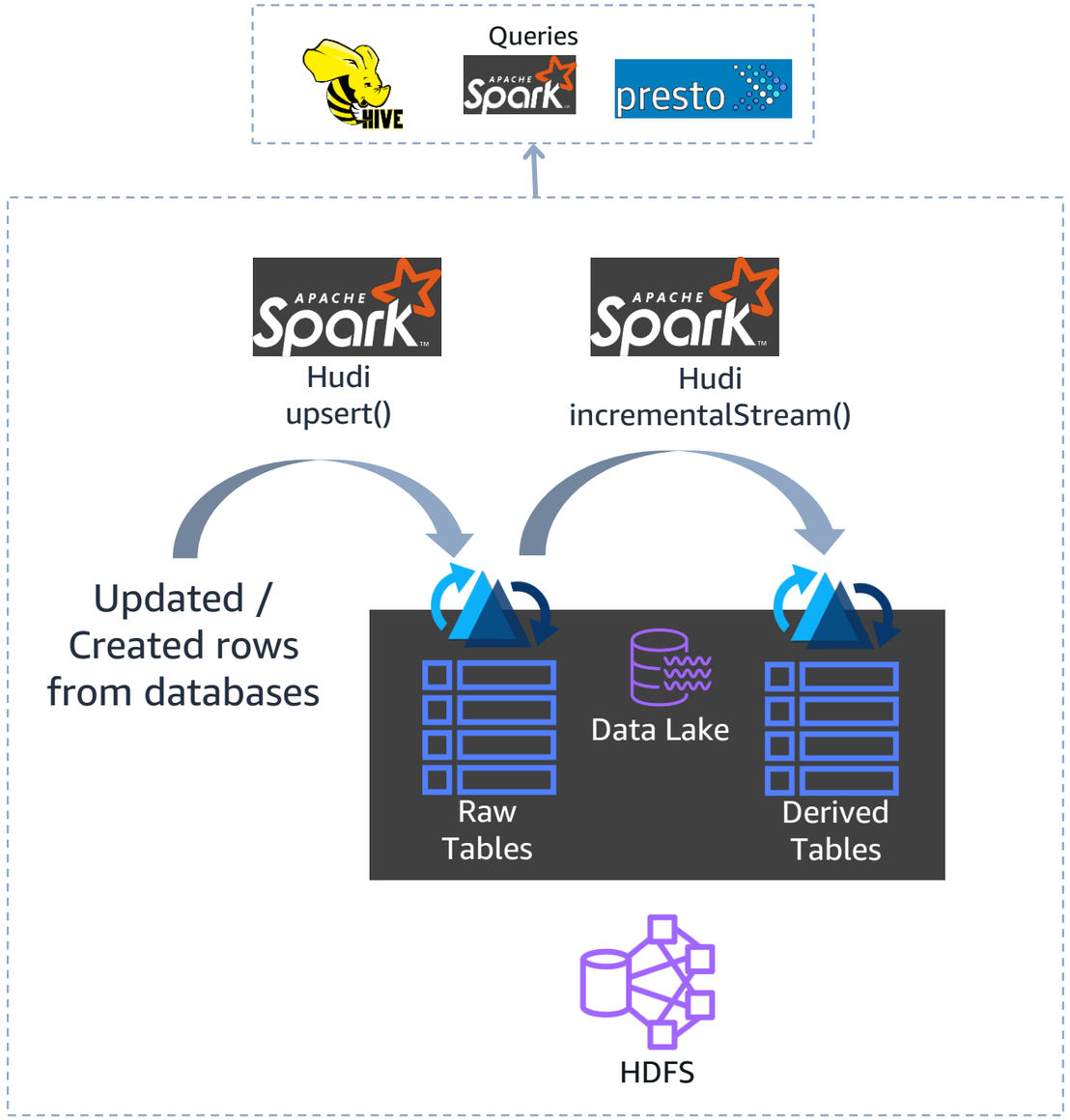
2018 : Picked up adopters, generalized for cloud

2019 : Incubated into ASF and working towards graduation

2020 : ASF announces Apache Hudi as a Top-Level project

Apache Hudi Data Lake

- Move away from big batches, and run mini-batches in a streaming fashion
- Think of this as a database problem
- Consider 10-100x more data scale & analytical workloads
- Give tuning options for different trade-offs



Why it matters?

- Near real-time data ingestion
- Supercharged batch jobs
- Stream processing on batch data
- Unified, optimized analytical storage
- Row-level deletions to simplify data privacy
- Building block for great data lakes!

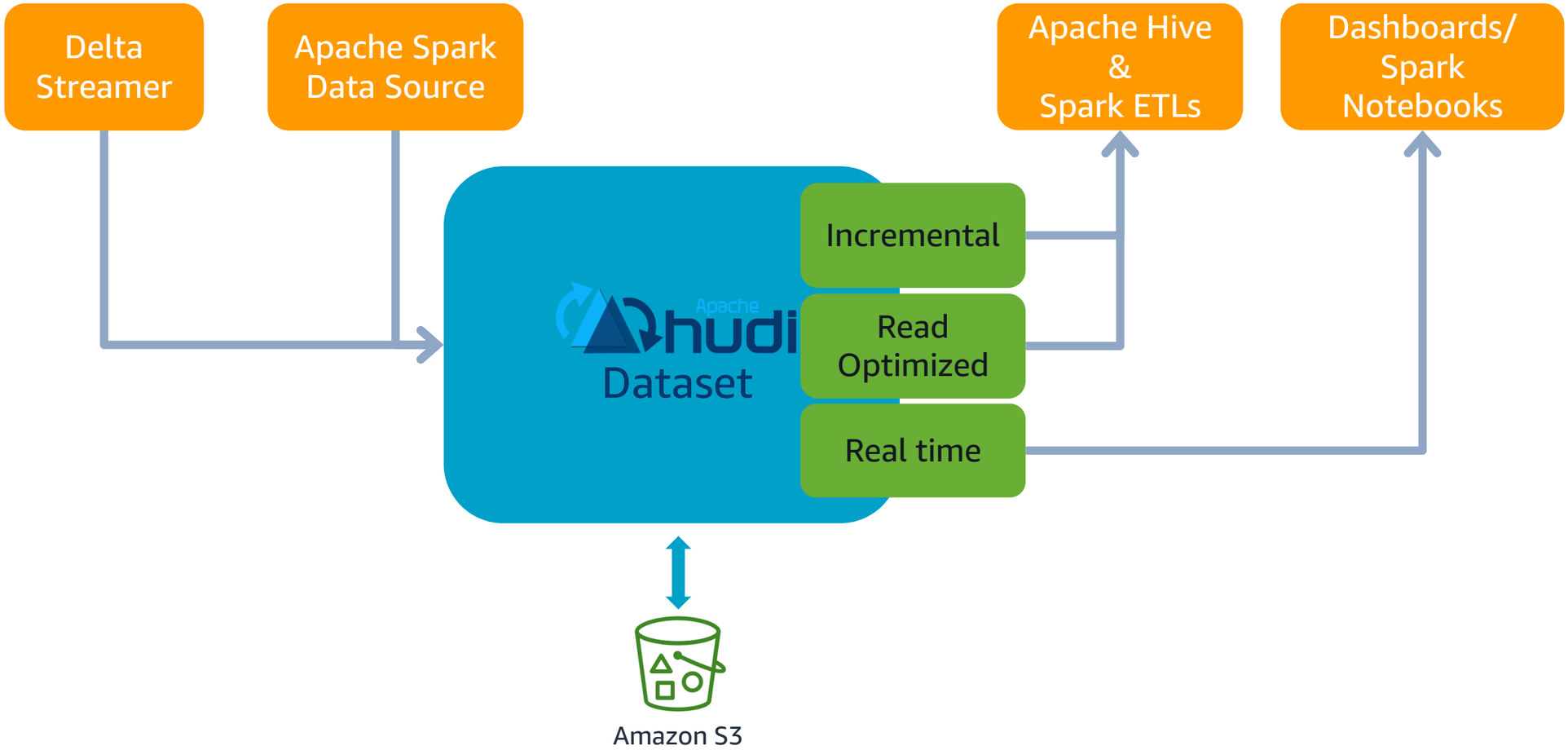
aws

Apache Hudi features

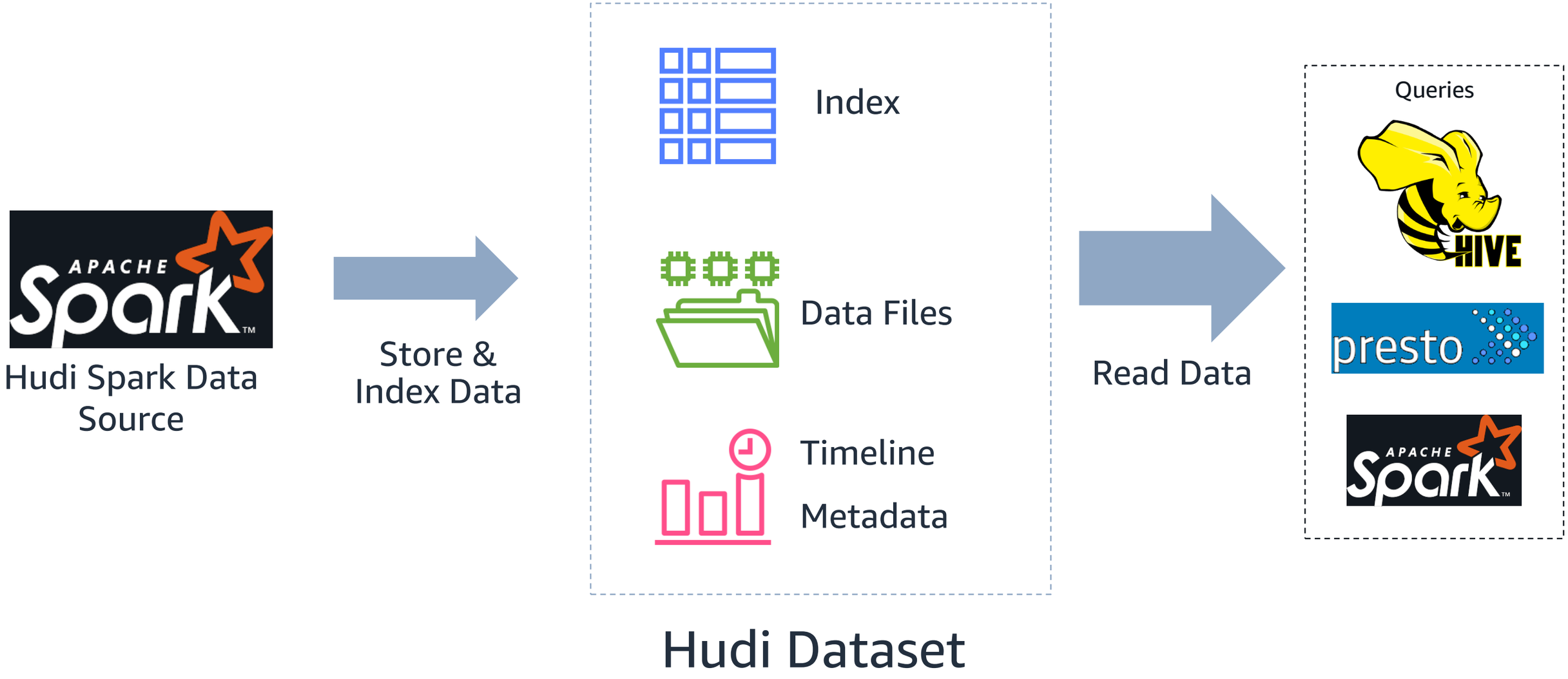
- Upsert support with fast, pluggable indexing
- Atomic publish with rollback, save points
- Snapshot isolation between writer & queries
- Manages file sizes, layout using statistics
- Async compaction of row & columnar data
- Timeline metadata to track lineage

aws

Apache Hudi Overview



Apache Hudi Architecture



Storage Types

Copy On Write

Read Heavy

Apache Hudi
Dataset

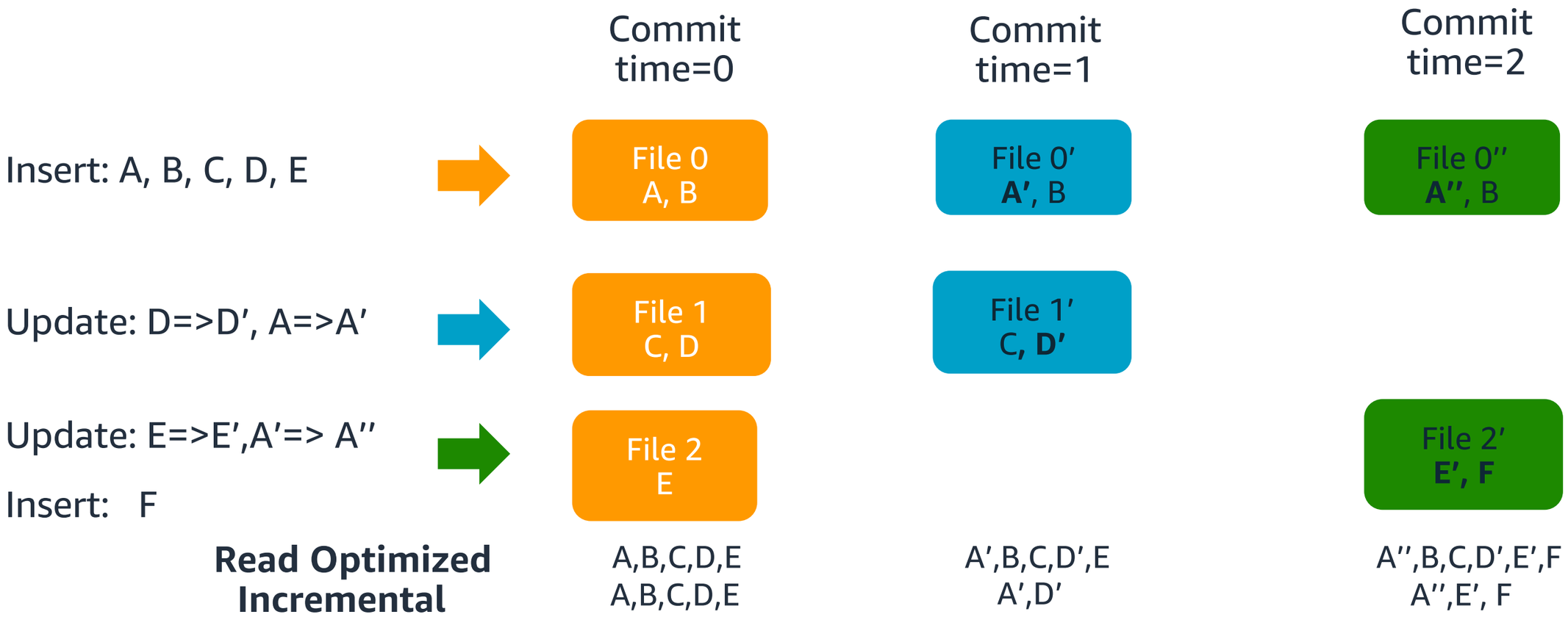
Merge On Read

Write Heavy

Storage Types & Views

Storage Type: Copy On Write

Views/Queries: Read Optimized, Incremental



Storage Types & Views

Storage Type: Copy On Write

Views: Read Optimized, Incremental

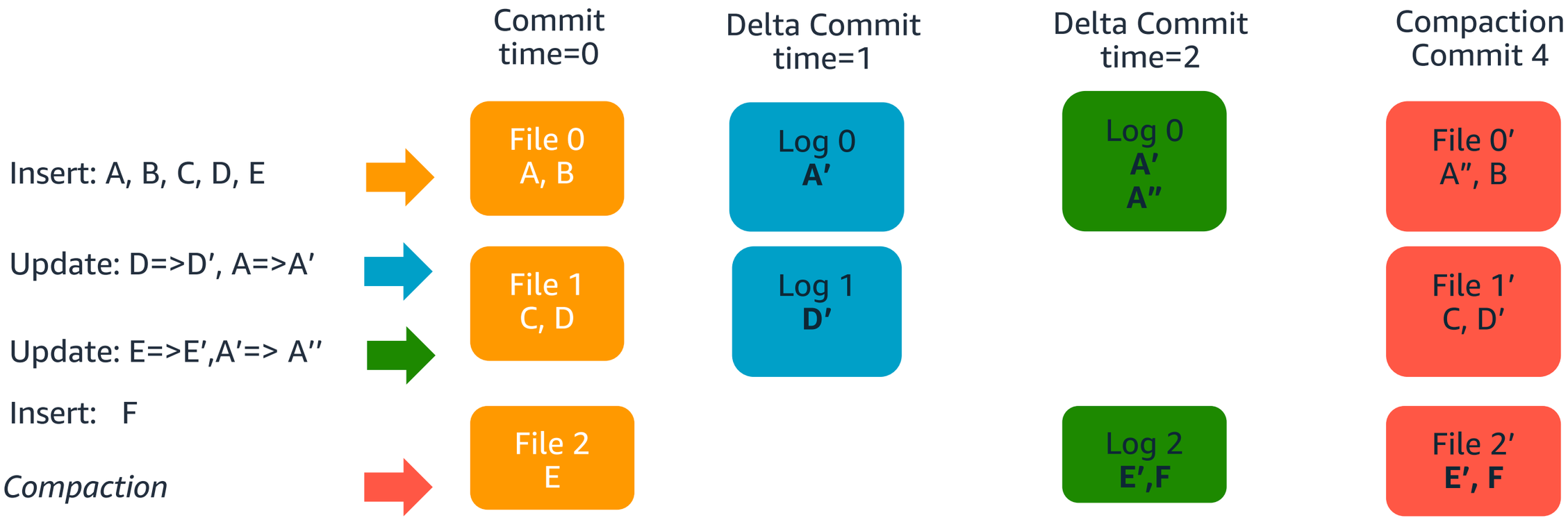
When to Use?

- Your current job is rewriting entire table/partition to deal with updates
- Your workload is fairly well understood and does not have sudden bursts
- You're already using Parquet files for your tables
- You want to keep things operationally simple

Storage Types & Views

Storage Type: Merge On Read

Views/Queries: Read Optimized, Incremental, Real Time



	Commit time=0	Delta Commit time=1	Delta Commit time=2	Compaction Commit 4
Real Time	A,B,C,D,E	A',B,C,D',E	A'',B,C,D',E',F	A'',B,C,D',E',F
Incremental	A,B,C,D,E	A',D'	A'',E', F	
Read Optimized	A,B,C,D,E	A,B,C,D,E	A,B,C,D,E	A'',B,C,D',E',F

Storage Types & Views

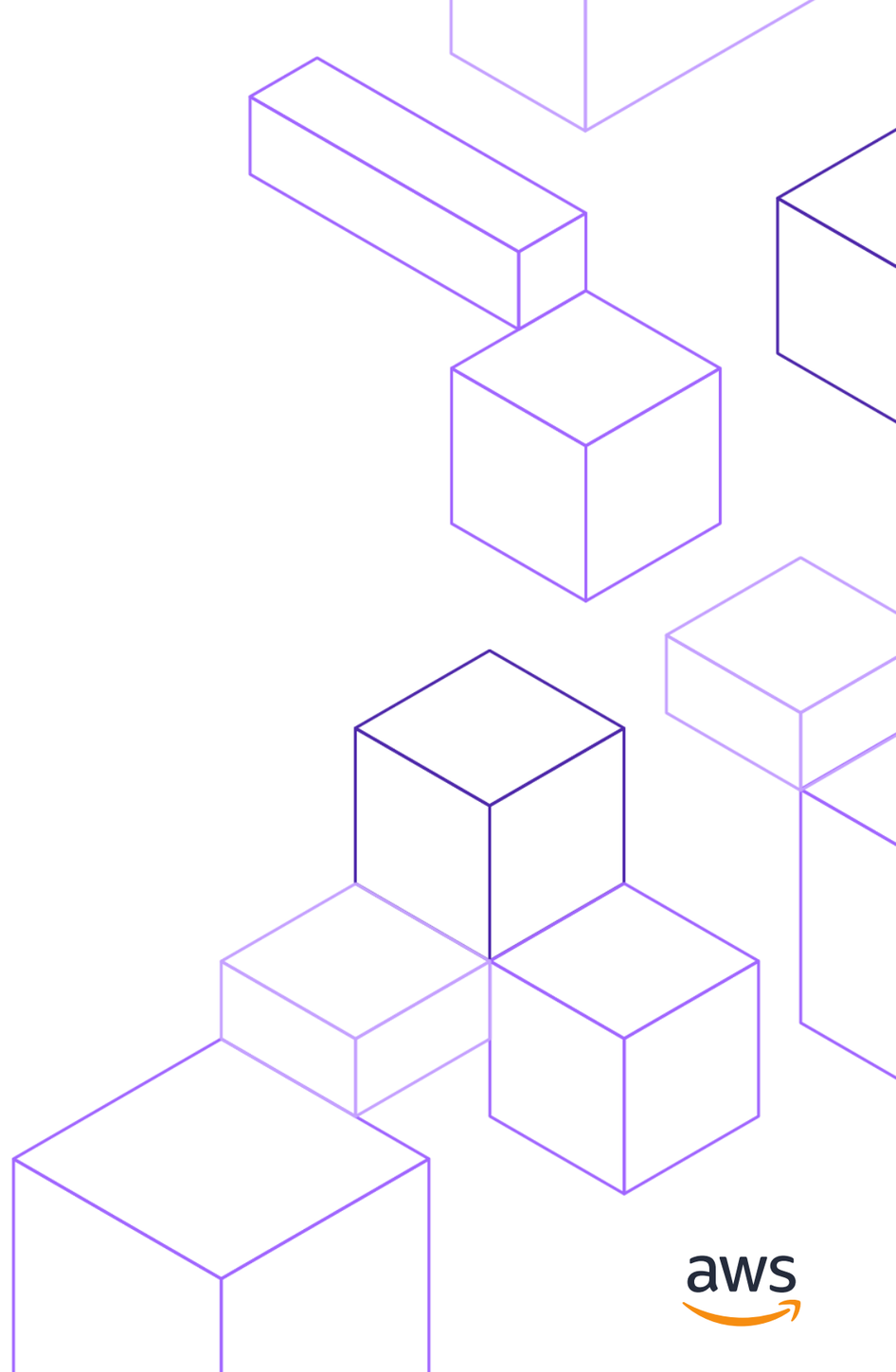
Storage Type: Merge On Read

Views: Read Optimized, Incremental, Real Time

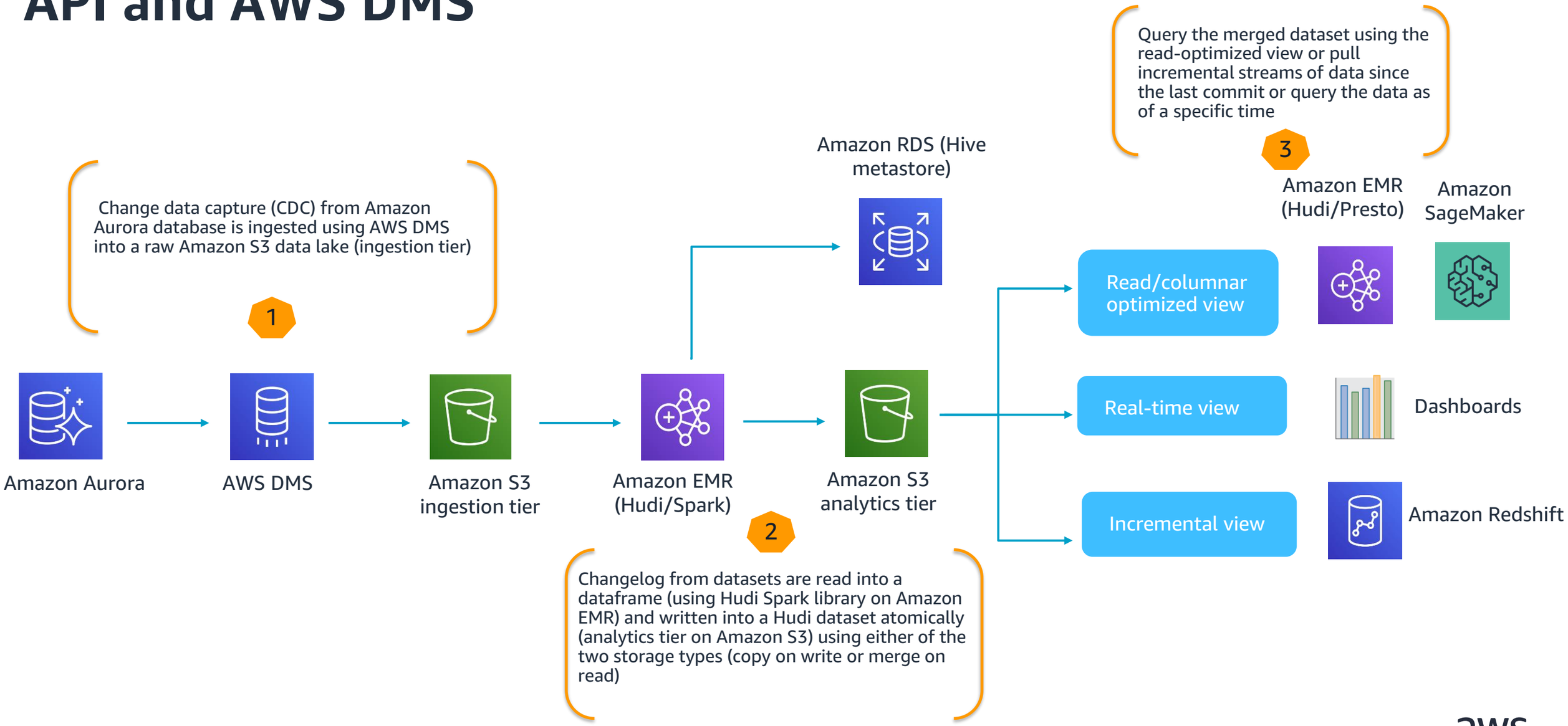
When to Use?

- You want ingested data available for query as fast as possible
- Your workload can have sudden spikes or changes in pattern
  - Example: bulk updates to older transactions in upstream database cause updates to old partitions in S3.

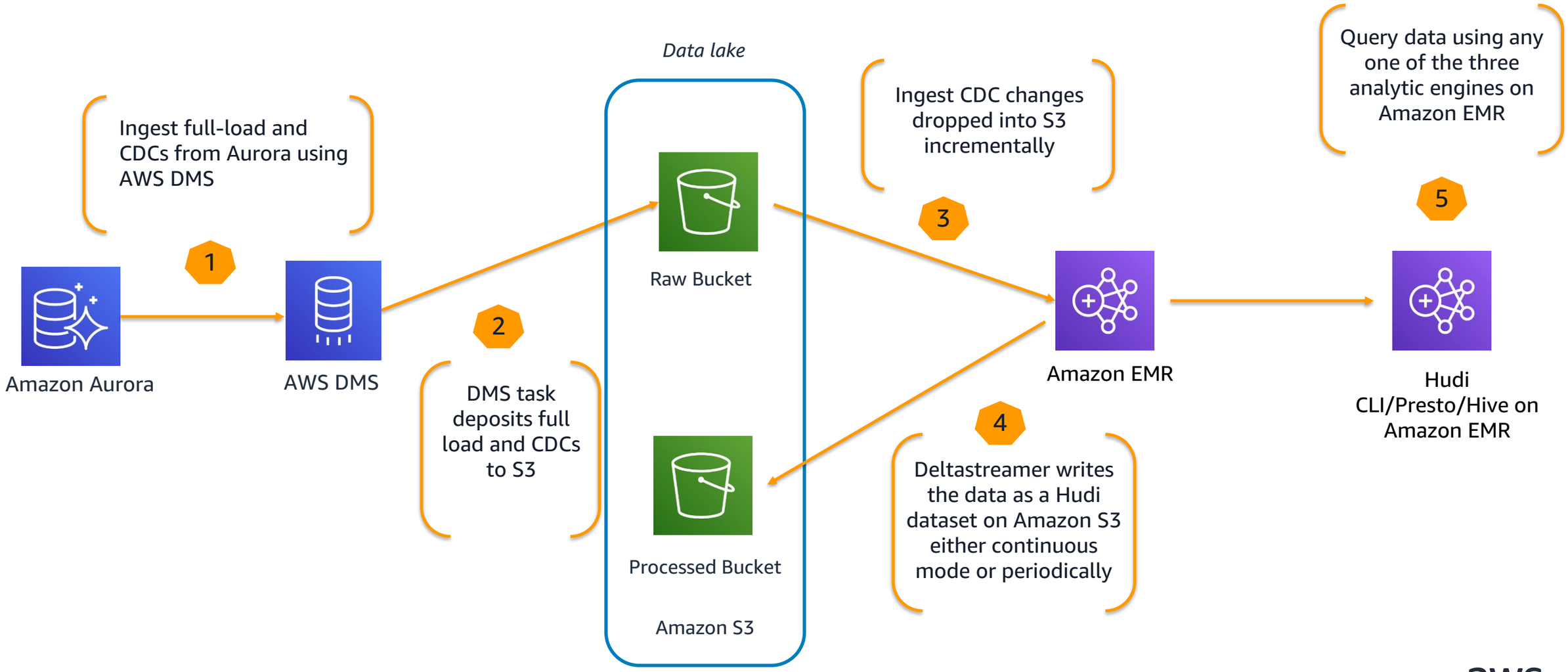
Architectural Patterns



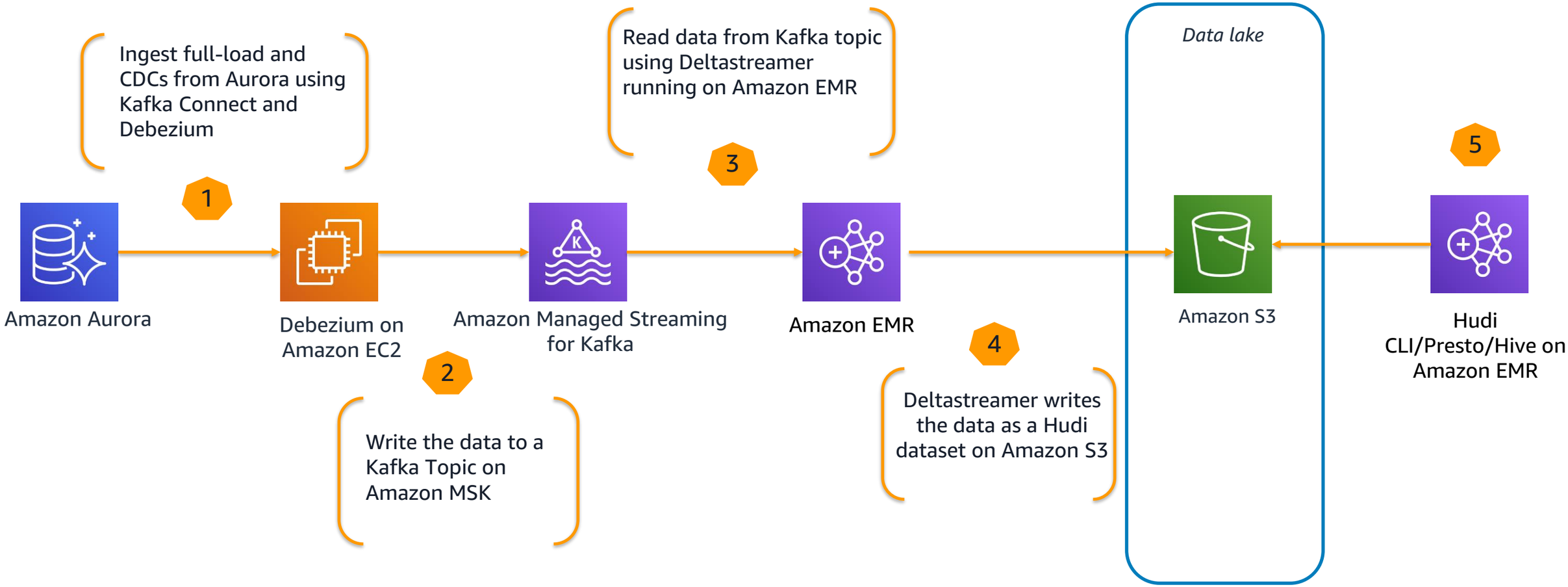
Using Apache Hudi Spark Datasource API and AWS DMS



Using Apache Hudi Deltastreamer and AWS DMS



Using Apache Hudi Deltastreamer and Amazon MSK



Recap

How can Apache Hudi help you?

- Data privacy law compliance
- Consuming real time data streams and applying CDC logs
- Reinstating late arriving data
- Tracking data changes and rollback
- Simplifying file management on Amazon S3

aws

Apache Hudi Resources

Demo Notebook: https://github.com/vasveena/Hudi_Demo_Notebook/blob/master/Hudi_Pyspark_Example-Copy1.ipynb

Workshop: https://incremental-data-processing-on-amazonemr.workshop.aws/en

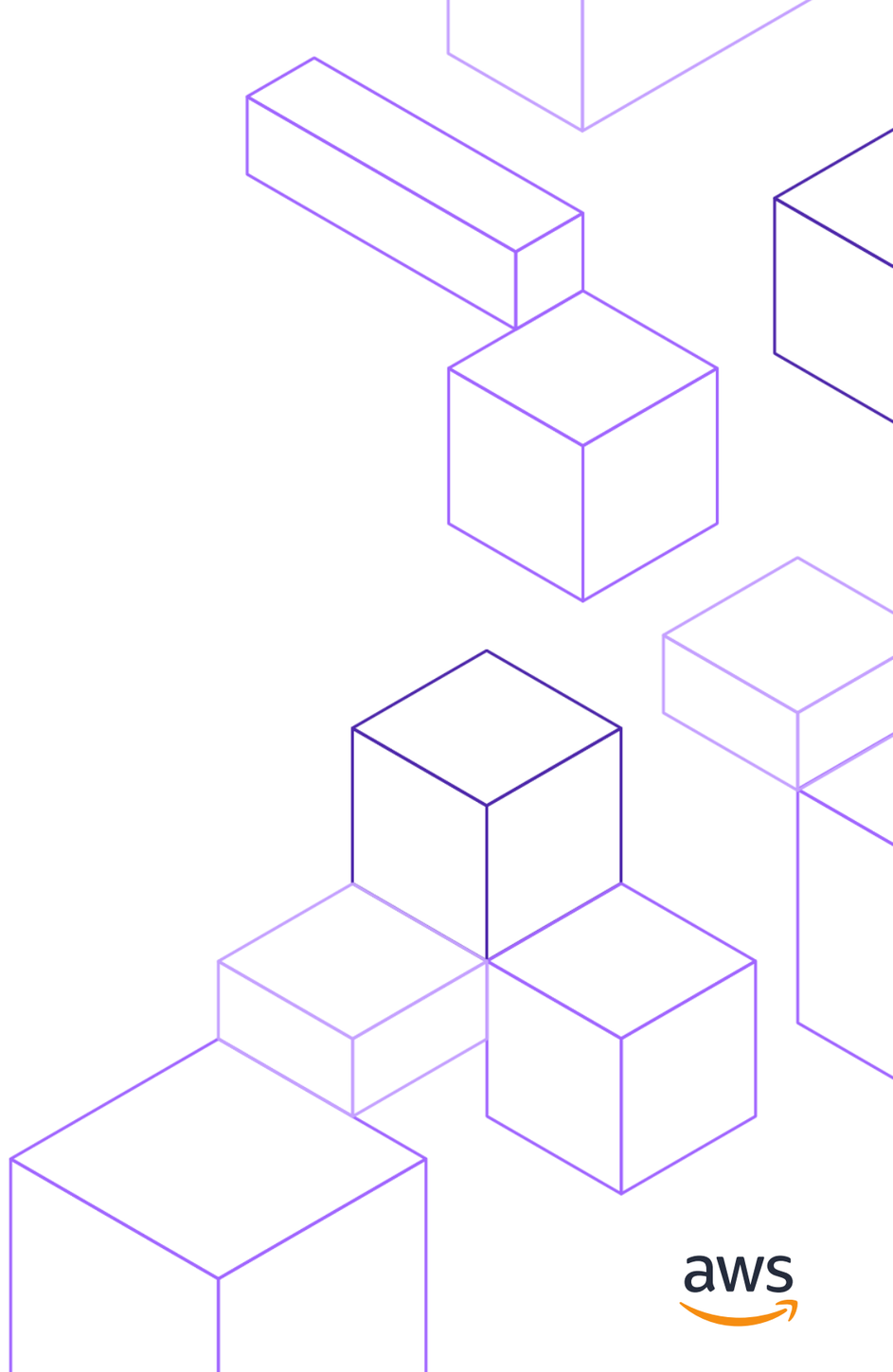
Documentation: https://hudi.apache.org

Github: https://github.com/apache/incubator-hudi/

FAQ: https://tinyurl.com/hudi-faq

aws

Apache Hudi Demo



Thank you !