Building a Cloud Data Platform with Delta Lake

Joshua Cook

**Fundamentals of Delta Lake**

In the previous lesson we discussed some of the challenges inherent to the cloud data platform architecture. These challenges become most apparent when trying to build AI, analytic, or machine learning projects using data from an EDSS built with a cloud data platform architecture.

It is possible to minimize these challenges by using Delta Lake to build a cloud data platform. In this lesson we will review fundamental concepts inherent to Delta Lake. We will also explore how Delta Lake enables organizations to avoid known challenges with data lakehouses to build powerful, next generation EDSSs.

**Common challenges faced when building a cloud data platform**

When building a traditional cloud data platform (not using Delta Lake), organizations typically face challenges related to reliability, performance, and engineering.

Reliability challenges

–

Reliability challenges can present themselves as:

* Poor quality or corrupt data due to failed writes or schema mismatch issues
* Inconsistent views when trying to read streaming data sources
* Storage issues including data lock-in and storage idiosyncrasies inherent to different on-premises or cloud environments

Performance challenges at scale

–

Performance challenges can present themselves as:

* A bottleneck in processing as readers struggle to deal with too many small or few massive files in a directory. Streaming compounds this problem.
* A breakdown of the indexing of tables using partitioning. Partitioning breaks down if the wrong fields are picked for or when data has many dimensions or high cardinality columns.
* A lack of caching can lead to low cloud storage throughput (cloud object storage is 20-50MB/s/core vs 300MB/s/core for local SSDs).

Engineering challenges associated with Lambda architecture

–

A lambda architecture, originally developed to enable the merging of batch and stream data sources, can require up to six times as many engineers to maintain as a delta architecture, for the same data pipeline.

**Delta Lake pillars**

Next, we will introduce the fundamental concepts behind Delta Lake and explain why building a cloud data platform with Delta Lake enables organizations to bypass these challenges.

Delta Lake is built upon three pillars addressing reliability, performance, and engineering challenges:

1. Clean, quality data
2. Consistent views across batch and stream data workloads
3. Optimized and easy to adopt

**Pillar 1: Clean, Quality Data**

Delta Lake provides high quality and reliable data that is always ready for analytics through a range of features for ingesting, managing, and cleaning data.

This pillar speaks to reliability challenges when building a cloud data platform.

Under this pillar, Delta Lake has the following features:

1. **ACID transactions** ensure that only complete writes are committed.
2. **Schema enforcement** automatically handles schema variations to prevent insertion of bad records during ingestion.
3. **Time Travel,** part of Delta Lake’s built-in data versioning, enables rollbacks, full historical audit trails, and reproducible machine learning experiments.
4. **Exactly once semantics** ensures that data are neither missed nor repeated erroneously.

**Pillar 2: Consistent Views Across Batch and Stream Data Workloads**

Delta Lake supports multiple simultaneous readers and writers for mixed batch and stream data.

This pillar speaks to performance challenges when building a cloud data platform.

Under this pillar, Delta Lake has the following features:

1. **Snapshot isolation** provides support for multiple simultaneous writers and readers.
2. **Mixed streaming and batch** data mean that a table in Delta Lake is a batch table as well as a streaming source and sink. Streaming data ingestion, batch historic backfill, and interactive queries all just work out of the box.

**Pillar 3: Optimized and easy to adopt**

Delta Lake is easy to adopt, optimized for the cloud, and using Delta Lake avoids data lock-in.

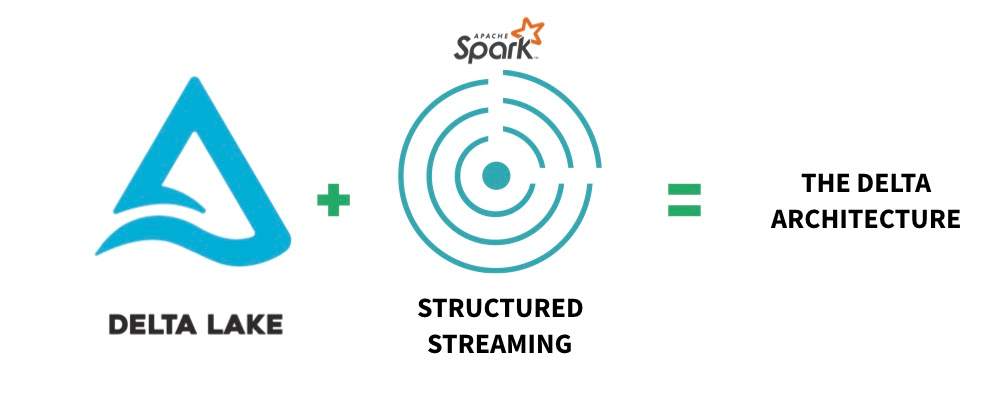
This pillar speaks to engineering challenges when building a cloud data platform.

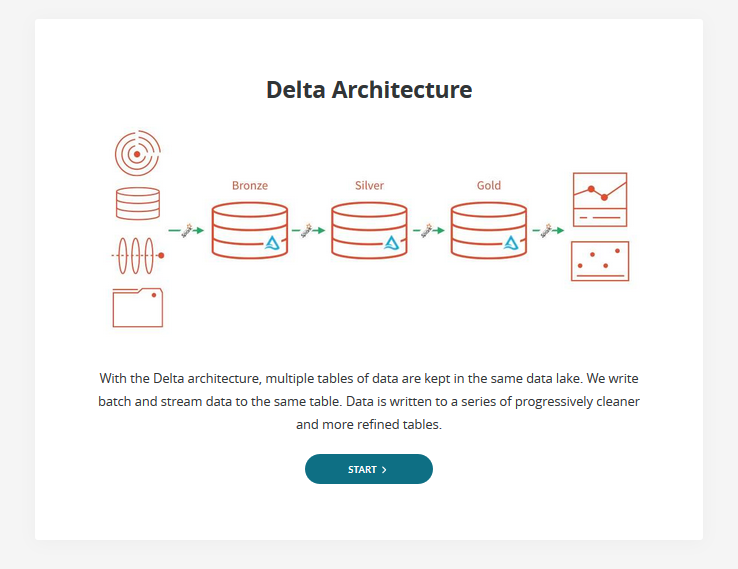
Under this pillar, Delta Lake has the following features:

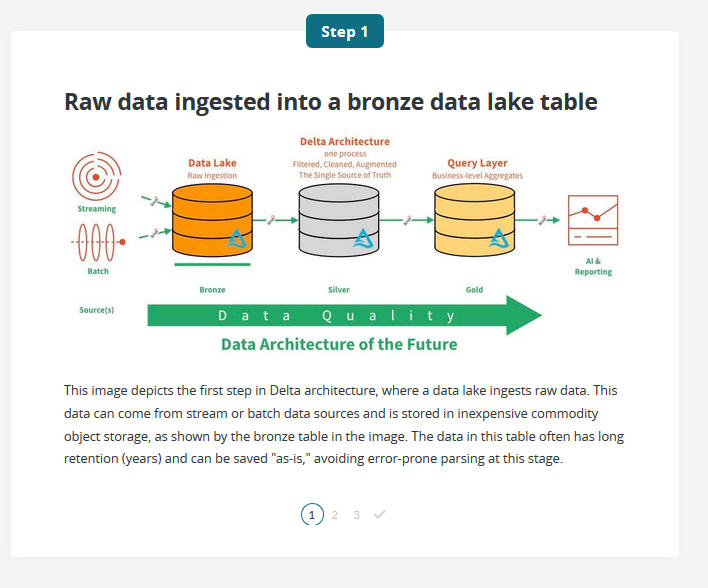
1. **Scalable metadata handling** leverages Spark's distributed processing power to handle all the metadata for petabyte-scale tables with billions of files at ease.
2. **Effective on-premises** means that Delta Lake works well with Hadoop Distributed File System (HDFS) on-premises.
3. **Compatibility with Spark APIs** means that Delta Lake is easy to adopt for Spark users.
4. As an **open-source format**, Delta Lake eliminates data lock-in. Using Delta Lake, there is no requirement only to use Delta Lake.
5. **Local development** means that Delta Lake supports laptop-based development and testing.
6. **In-place import** allows efficient, fast-import from Parquet to Delta format.

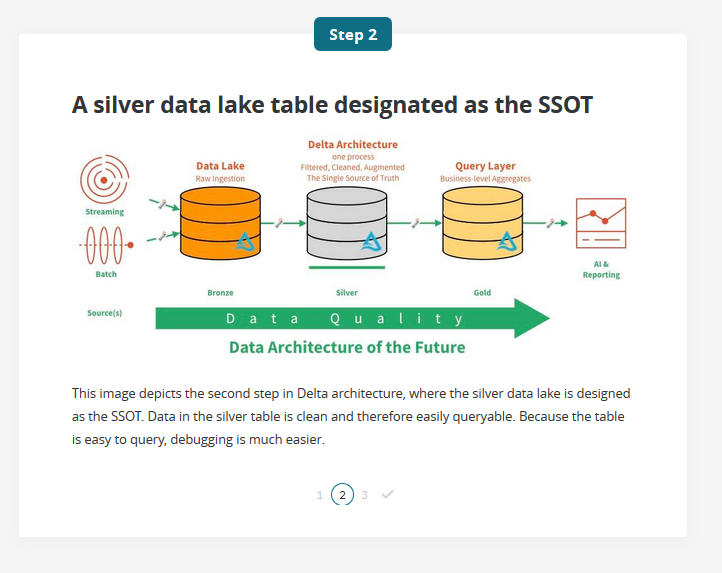
**Exploring the Delta Architecture**

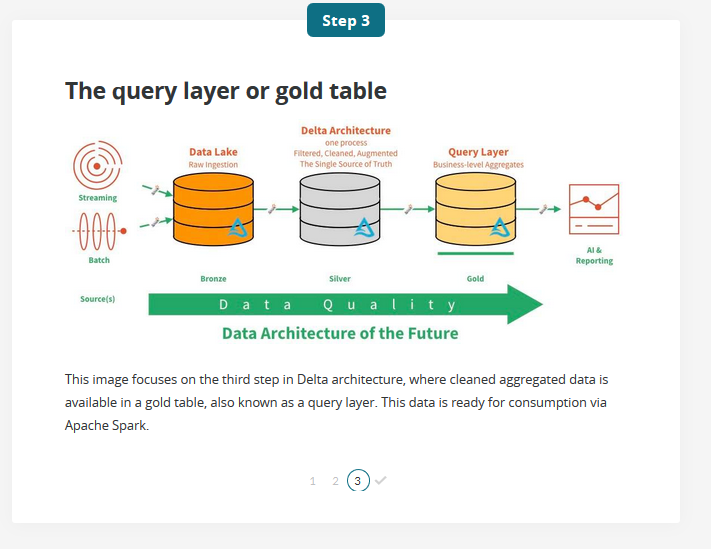
Beyond these core features, using Delta Lake to build a cloud data platform also allows the use of Delta architecture. Let’s explore this architecture and its advantages when building an EDSS.











**A Complete Architecture That Fully Supports Batch and Stream Data Processing**

Delta architecture is a complete architecture that fully supports batch and stream data processing. In this image, we can see data streams moving through a data pipeline. These streams have low latency or are manually triggered, which eliminates the need for schedules or jobs. In addition, this architecture supports methods not historically supported by an ODW or CDW such as **DELETE**, **MERGE**, **UPDATE**, and Upserts, useful for GDPR or CCPA compliance.

