CLOUDERA

WHITE PAPER

DATA ARCHITECTURE SERIES

THE OPEN DATA LAKEHOUSE

Building a Modern Data Lakehouse with Cloudera

Abstract

This whitepaper provides an introduction to the Data Lakehouse architecture. It explains what it is, why it was created, the challenges it addresses, offers a Cloudera-based reference architecture and highlights two key areas the Lakehouse can be extended.

Version: 1.0
Author: Daniel J Hand

Table of Contents

Introduction

In this section we briefly summarise why Cloudera wrote this whitepaper, who it is intended for, why they should read it and recommendations for further reading.

Audience

This whitepaper was written for members of Architecture, Operations, Engineering and Business leaders of Enterprise Data Platform teams. It may also provide useful reading for Chief Data Officers (CDO) and Chief Information Officers (CIO) that want to establish or strengthen their understanding of the Data Lakehouse architecture, specifically as it applies to Cloudera's products and services.

Purpose

The Data Lakehouse is one of three important emerging data architectures; the other two are Data Mesh and Data Fabric. Organisations need to clearly understand what each of them is, why they are important and how to implement them at scale, in a hybrid landscape.

Cloudera has been helping organisations implement Data Lakehouse architectures for several years. With the recent introduction of multiple analytical services as cloud-native Data Services and a new table storage format, we can now fully support key management features of Data Warehouses, such as transactions, data/table versioning and snap-shots.

Recommended Reading

The recommended reading listed below is limited to only those sources that directly support this whitepaper.

- Official Cloudera Blog
- Exploring Lakehouse Architecture & use cases, Gartner Research, Jan 2022
- Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics
- Introducing Apache Iceberg in Cloudera Data Platform
- 5 Reasons to Use Apache Iceberg on Cloudera Data Platform (CDP)
- A comparison of Data Lake table formats

What Is The Data Lakehouse Architecture

In this section we introduce the Data Lakehouse architecture. We consider its origins, the challenges it addresses, commonly understood, but evolving definitions and areas for improvement.

Definition

Gartner defines the Data Lakehouse architecture or paradigm as:

> "Data Lakehouses integrate and unify the capabilities of Data Warehouses and Data Lakes, aiming to support AI, BI, ML and Data Engineering on a single platform."
>
> Exploring Lakehouse architecture & use cases, Jan 2022

This definition has expanded over time to accommodate more analytical services. Cloudera expects this trend to continue in the future and include scope for real-time streaming analytics and operational data stores.

Origin

The term Data Lakehouse first entered the Enterprise Data Platform lexicon back in 2017. It was used to describe how Jellyvision had combined structured data processing (Data Warehouse) with a schemaless system (Data Lake). Combining together these two architectural paradigms, led to the Data Lakehouse.

Since then, the term's definition has evolved to include additional analytical services such as Machine Learning (ML), but also greater support for the management features of traditional Data Warehouses.

Qualities

A modern Data Lakehouse should bring together the benefits of a Data Lake and a Data Warehouse at a low TCO. It should therefore possess the following key qualities:

- Open, flexible and performant file and table formats e.g. Apache Parquet, Iceberg
- ACID Transactions, table versioning, snap-shots and sharing at petabyte scale
- Multifunction analytics across an open ecosystem
- Strong data management (security, governance & lineage)
- Strong data quality and reliability
- Best in class SQL performance
- Direct and declarative access for non SQL interactions

Why The Data Lakehouse Architecture Is Useful

In this section we consider why the Data Lakehouse architecture is useful. We consider the limitations of the Data Lake and Data Warehouse and see how Data Lakehouse overcomes these limitations while maintaining the qualities of both.

Limitations of Data Lake and Data Warehouse

To understand why the Data Lakehouse architecture is growing in popularity, we need to consider the architectures it replaces and their limitations. The Data Lakehouse architecture replaces the largely independent Data Lake and Data Warehouse architectures.

Data is first ingested into a Data Lake by an ETL operation from each source system. Historically, these sources would mainly be operational systems containing structured data. However, today more than half of the data ingested is semi-structured or unstructured data.

Data is then loaded into a Data Warehouse with another ETL operation. Data is conformed into a given logical data model, often on an underlying proprietary storage layer. SQL can then be used to query the data and we get to benefit from DBMS features of the Data Warehouse such as support for transactions, table versioning and snap-shots.

While this architecture provides the economic benefits of cheap, scalable storage in the Data Lake, it suffers from three main challenges.

- Data duplication: Multiple copies of data are required. Once data is copied from the source systems to the Data Lake, it's copied again from the Data Lake to the Data Warehouse. A partial solution to this problem is to use external tables, at the expense of reduced management, in particular support for ACID transactions. Data duplication leads to three challenges:
  - Data staleness—Data in the Data Warehouse is almost always out of sync with data in the Data Lake, which itself is out of sync with data in each source system.
  - Data quality & reliability—Multiple ETL operations getting data from source systems to the Data Warehouse increases the likelihood of failure. Inconsistencies between different processing engines may impact quality.
  - Increased cost—Intermediate storage and repeated ETL operations consume additional storage and processing cycles respectively.
- Limited support for analytical services: Data Warehouses were designed for Business Intelligence (BI), reporting and Advanced Analytics. They lack support for ML with open frameworks and libraries. ML typically requires processing large amounts of structured and unstructured data so only providing an SQL interface is inefficient. Direct access to the underlying storage while taking advantage of previously defined schemas to better support working with DataFrames was missing. Support for real-time analytics, operational data stores and some categories of data such as time-series, are also generally poorly supported by traditional Data Warehouses. This leads to the purchase and operation of multiple analytical systems, each suffering from its own data duplication issues.
- Flexibility: Traditional Data Warehouses predominately run on premises on proprietary hardware. Cloud options are limited. Additional capacity needs to be purchased in large increments making scaling both inefficient and requiring significant lead time.

Each of the three major limitations results in complexity, reliability, quality and increased TCO.

Overcoming Limitations While Delivering Qualities

The Data Lakehouse architecture addresses these limitations while meeting the qualities in the following ways.

Avoiding Data Duplication

Instead of data being copied from source systems into a Data Lake and then again into a Data Warehouse, the Data Lakehouse provides a layer of abstraction to the underlying data in the Lake. This transactional metadata layer on top of the underlying Data Lake provides support for common Data Warehouse management features such as transactions, data versioning and snap-shots. Reducing the number of ETL steps to get data into the Data Warehouse reduces the likelihood of errors, improved efficiency and potential inconsistencies in processing engines.

Supporting an Open Ecosystem of Analytical Engines

Providing an efficient SQL interface for BI and reporting is necessary but insufficient and quite limiting when supporting ML. Systems that implement the Data Lakehouse architecture therefore need to be able to provide direct access to the underlying data in the lake. At the same time, ML frameworks must be able to take advantage of metadata to simplify the process of importing data into DataFrames for Data Science pipeline building and model creation.

Flexible Hybrid Deployment Options

In order to significantly reduce the TCO, we must move away from expensive and proprietary hardware. Modern implementations of the Data Lakehouse architecture decouple compute and storage and opt for cloud-native architectures. This allows each to scale independently and simplifies running multiple analytical workloads across shared data.

Data Lakehouses are required on premises on commodity hardware and in the public cloud. There are advantages for adopting cloud native hybrid solutions that can leverage object storage and managed container services across each environment.

Building an Open Data Lakehouse

In this section we provide an introduction to the Cloudera Data Platform (CDP), with a focus on CDP Public Cloud. We then summarise the key logical service components that support Cloudera's Open Data Lakehouse. We describe how Apache Iceberg provides a flexible, scalable table format to support schema-based access to multiple analytical services across the data lifecycle. We conclude by looking beyond the Data Lakehouse as we know it today and share how Cloudera is bringing streaming analytics into the Data Lakehouse.

The Cloudera Data Platform (CDP)

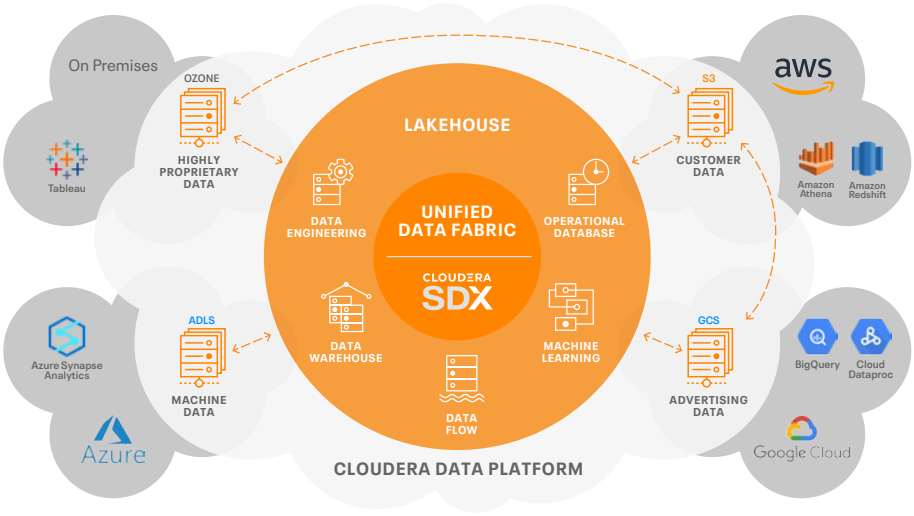


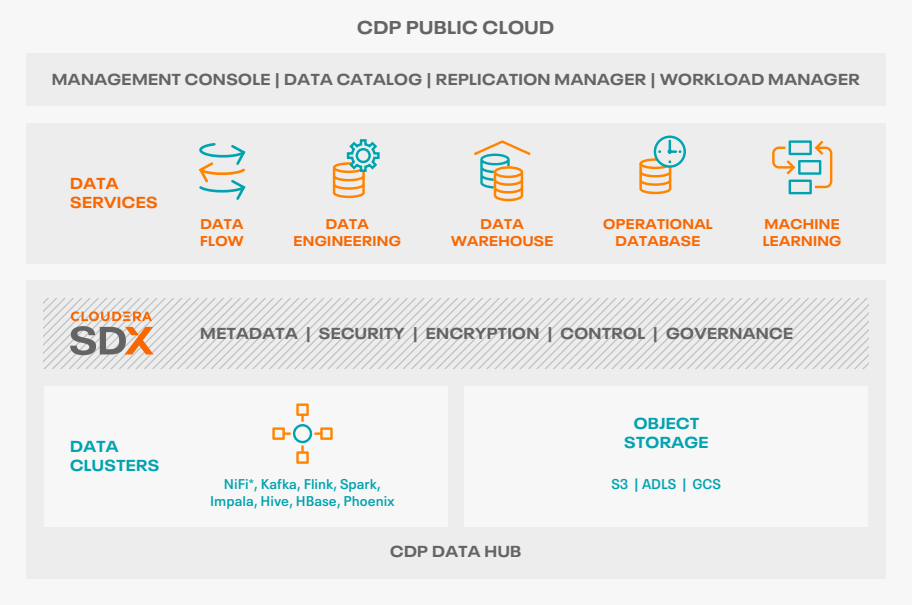
Figure 01—The Cloudera Data Platform (CDP)

The Cloudera Data Platform (CDP) is a hybrid data platform designed to provide the freedom to choose any cloud, any analytics and any data. CDP delivers fast and easy data management and data analytics for data anywhere, with optimal performance, scalability and security.

CDP provides the freedom to securely move applications, data and users bi-directionally between data centres and multiple data clouds, regardless of where data resides. This is made possible by embracing three modern data architectures:

- An Open Data Lakehouse enables multi-function analytics on both streaming and stored data in a cloud-native object store across hybrid and multi-cloud
- A unified Data Fabric centrally orchestrates disparate data sources intelligently and securely across multiple clouds and on premises
- A scalable Data Mesh helps eliminate data silos by distributing ownership to cross-functional teams while maintaining a common data infrastructure

Figure 02 provides a summary of the service components that make up CDP Public Cloud. We'll now explore how each of these components supports the Data Lakehouse architecture.



*Flow Management rate card

Figure 02—Services Components of CDP Public Cloud

Shared Data Experience (SDX)

Cloudera Shared Data Experience (SDX) combines enterprise-grade centralised security, governance, lineage and management capabilities with shared metadata and a data catalog. It provides a governance layer around cloud native object storage to deliver a Data Lake.

Data Hub

Data Hub allows users to deploy analytical clusters across the entire data lifecycle as elastic IaaS experiences. It provides the greatest control over cluster configurations, including hardware and individual service components installed. Its cloud native design supports separation of compute and storage with the unit of compute being a virtual machine. It provides support for auto scaling of resources based on environmental triggers.

Data Services

Data Services are containerised analytical applications that scale dynamically and can be upgraded independently. Through the use of containers deployed on cloud managed Kubernetes services such as Amazon EKS, Microsoft Azure AKS and Google GKE, users are able to deploy similar clusters to what is possible in Data Hub but with the added advantage of them being delivered as a PaaS experience. Cloudera Data Flow (CDF), Cloudera Data Engineering (CDE), Cloudera Data Warehousing (CDW), Cloudera Operational Database (COD) and Cloudera Machine Learning (CML) are all available as Data Services on CDP Public Cloud.

Cloudera Data Engineering (CDE)

CDP Data Engineering (CDE) is a cloud-native data engineering Data Service. Building on Apache Spark, CDE enables orchestration automation with Apache Airflow, advanced pipeline monitoring, visual troubleshooting and comprehensive management tools to streamline ETL processes across enterprise analytics teams.

CDE is fully integrated with CDP, enabling end-to-end visibility, security and data lineage with SDX as well as seamless integrations with other CDP services such as Data Warehouse and Machine Learning.

Cloudera Data Warehouse (CDW)

While it is possible to achieve many of the qualities of a traditional Data Warehouse using a combination of Apache HIVE or HIVE ACID together with the HIVE table format, the combination of Apache Impala and Apache Iceberg provides broader coverage. We therefore recommend Apache Impala as the transactional Data Warehouse engine for your Data Lakehouse.

Today, we support storing and querying Iceberg tables. Support for ACID transactions will be available in August, 2022. The Hive metastore stores Iceberg metadata, which includes the location of the table on the Data Lake. However, unlike the HIVE table format, Iceberg stores both the data and metadata on the Data Lake leading to a number of advantages as we'll see in a later section.

Cloudera Machine Learning (CML)

Cloudera Machine Learning (CML) is a machine learning workflow solution that supports the entire Data Science lifecycle. Similar to CDW, it's designed to use containers for efficient data engineering and machine learning tasks. It provides support for the python and R programming languages and commonly uses open source machine learning libraries and frameworks.

CML supports experimentation and scoring on ML model pipelines to systematically select the best ML algorithm and tune model parameters. Once trained, ML models can be deployed and managed behind a protected RESTful API.

ML model performance can be monitored overtime to detect model drift. If performance drops below a threshold level, retraining and redeployment of the model can be automatically scheduled.

Accessing Iceberg tables from CML is simple and intuitive. Using the Spark engine, we create a connection that includes the Iceberg `spark-runtime, iceberg-session and pluggable-spark-session-catalog` jars. We specify the location of the database catalog file and specify the type to be 'hive'. We are now ready to interact with the database using Spark SQL.

Data Catalog

The Data Catalog provides a centralised and scalable way to democratise access to data across the Data Lakehouse. It helps answer questions such as "what data do we have?", "Where is it located?" and "Who owns it?". It also provides data profiling, data lineage, security and classification and audit features.

Management Console

The Management Console provides a single pane of glass to manage CDP Public Cloud, CDP Private Cloud and legacy versions of CDH and HDP. It supports the administration of users, environments and analytical services supporting each Data Lakehouse.

Apache Iceberg—An Open Table Format

Apache Iceberg is an open source project within the Apache foundation. Open sourced by Netflix in 2018, it has since grown to be a leading open table format with a strong community of contributors; they include Tabular, Apple, Netflix, LinkedIn, multiple public cloud vendors and of course Cloudera. Collectively, this community is ensuring rapid innovation within Iceberg but also a commitment to open standards and therefore an open ecosystem. Today, Iceberg supports the broadest range of operations by third-party engines.

As highlighted earlier in the document, a Data Lakehouse possesses a set of qualities. Those qualities are the union of those from a Data Lake and those from a Data Warehouse. We cannot simply bring together a processing engine and a flexible table format, and say it implements a Data Lakehouse. We must also integrate the qualities of a Data Lake.

Iceberg provides a flexible and open storage format that supports petabyte scale tables. As illustrated in figure 03, It does this by storing both the data and metadata in the Data Lake. Data is typically stored in Apache Parquet format and the associated metadata in Apache Avro format. Entries in the Data Catalog are then a pointer to the manifest file on the Data Lake.

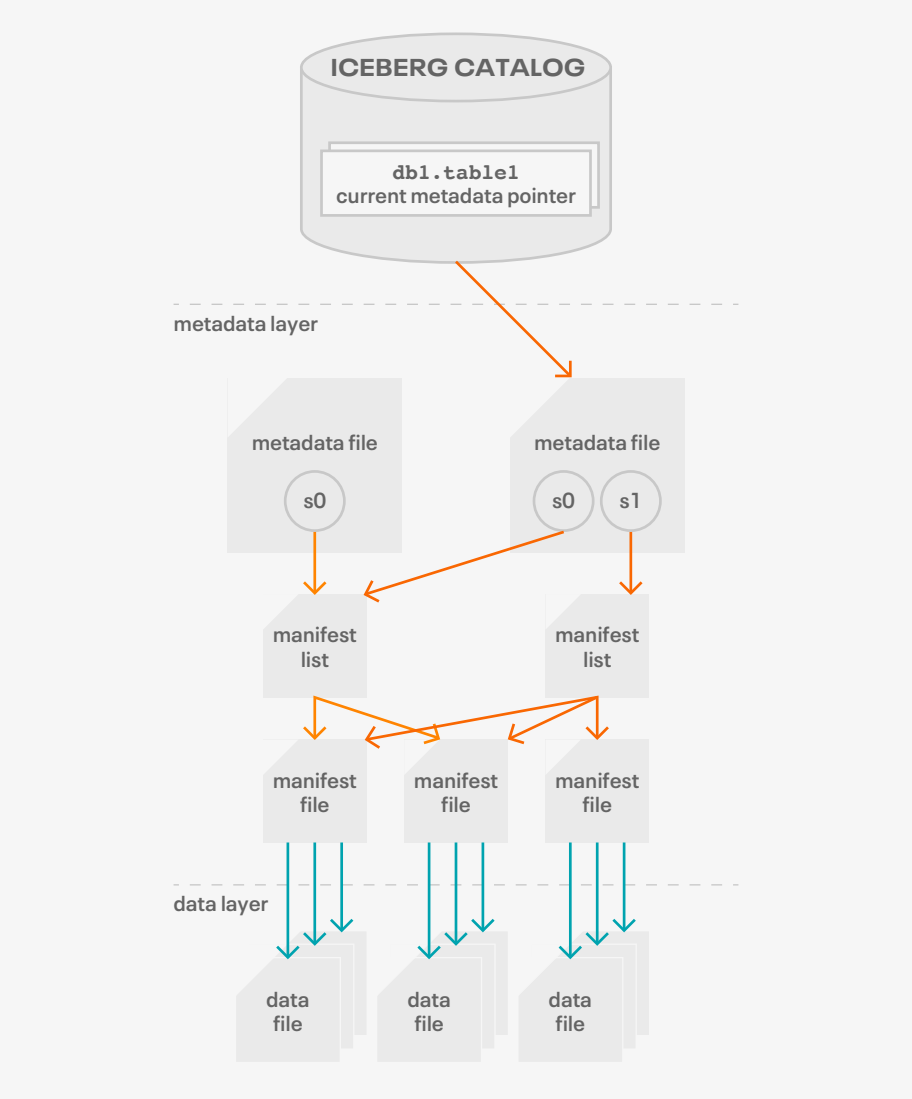


Figure 03— Apache Iceberg Table Architecture

Iceberg also supports many of the management features of a traditional Data Warehouse. These include transactions, data versioning and snap-shots. Iceberg supports flexible SQL commands to merge new data, update existing rows, and perform targeted deletes. Time-travel enables reproducible queries that use exactly the same table snapshot, or lets users easily examine changes. Version rollback allows users to quickly correct problems by resetting tables to a previously known state.

Support for Iceberg in our Data Services on CDP Public Cloud became Generally Available in June, 2022. It will be available in CDP Private Cloud shortly thereafter.

About Cloudera
At Cloudera, we believe that data can make what is impossible today, possible tomorrow. We empower people to transform complex data into clear and actionable insights. Cloudera delivers an enterprise data cloud for any data, anywhere, from the Edge to AI. Powered by the relentless innovation of the open source community, Cloudera advances digital transformation for the world's largest enterprises.

Learn more at cloudera.com

Connect with Cloudera
About Cloudera:
cloudera.com/more/about.html

Read our Blog:
blog.cloudera.com

Follow us on Twitter:
twitter.com/cloudera

Visit us on Facebook:
facebook.com/cloudera

See us on YouTube:
youtube.com/c/ClouderaInc

Join the Cloudera Community:
community.cloudera.com

Read about our customers' successes:
cloudera.com/more/customers.html

Data Quality

In a traditional Data Warehouse, data typically goes through three distinct stages, resulting in data of increasingly greater quality. These stages are commonly referred to as Landing, Refined and Production or Bronze, Silver and Gold. In the Landing stage, data is in its raw or natural format e.g csv format. As we transform and curate the data, we change its format e.g. Parquet, apply Data Modelling and store data in an Iceberg table in preparation for efficient analytics. This transformation results in data transitioning to the Refined stage. The final transition from Refined to Production requires data to be optimised for production usage. This may include data cleansing and normalisation operations

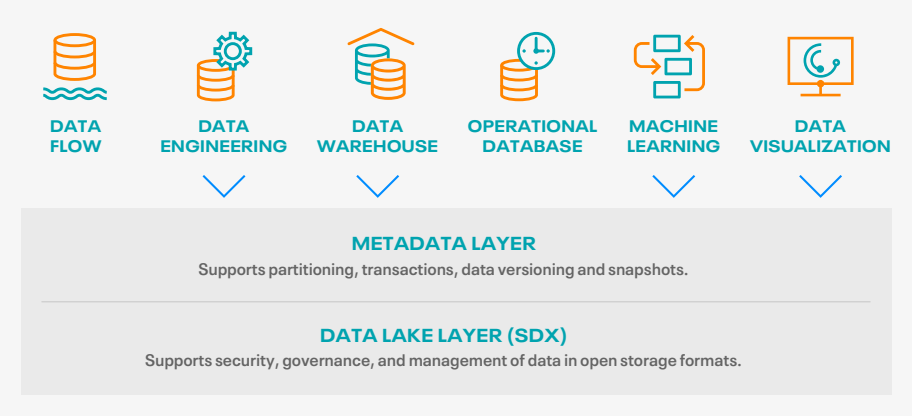


Figure 04 - A Simplified Systems View of the Data Lakehouse

We include the Iceberg client library in Cloudera's Data Services. This makes it possible to execute the transformations to move data between each of the three stages of quality. As Iceberg is open source, it's also readily available to integrate with third-party products and services to perform data quality operations.

Beyond The Data Lakehouse

As previously described, the definition of a Data Lakehouse has steadily evolved from originally supporting BI on a Data Lake, to today, supporting AI, BI, ML and Data Engineering on a single platform. In the earlier section "What is a Data Lakehouse Architecture" we introduced seven qualities that all Data Lakehouses share. One quality that we believe can be extended further, is to include support for additional analytical services. As such, we are working hard to extend the supported analytical services to include real-time analytics and operational datastores.

At Cloudera, we believe that an Open Data Lakehouse needs to extend beyond supporting a single processing engine. Today, we support Iceberg with Apache Spark, Apache Hive and Apache Impala. Collectively they support the Data Lakehouse architecture across Data Engineering, Data Warehousing and Machine Learning. Looking to the future, we will bring support to the real-time analytics engines Apache Flink, data flow management engine Apache Nifi and operational data stores powered by Apache HBase. This will provide the foundation of the next generation of Data Lakehouse, one that encompasses the entire data lifecycle—from the edge to AI.

Cloudera, Inc. 5470 Great America Pkwy, Santa Clara, CA 95054 USA cloudera.com