Delta Sharing: Multi-Cloud Data Sharing for Architects

Ryan Chynoweth

6 min read · Nov 13, 2023

Introduction

Organizations are leveraging multiple cloud hyperscalers to cater to their ever-evolving business needs. This approach enables them to effectively distribute workloads across cloud providers. Data is often stored where it is generated, however, the location of data is not always where the data is needed. Therefore, the imperative of seamlessly sharing data across various cloud environments, solutions, and on-premises storage is a strong requirement for all organizations.

The ability to facilitate such cross-cloud, hybrid-cloud, cross-locality data sharing has emerged as a mission-critical need for modern businesses. Within Databricks, we will dive into three distinct methodologies for achieving efficient data sharing:

- Delta Sharing

- Table Clones
- Spark Read and Write methods

Why Data Sharing is Important

In the past, data sharing within and beyond organizations involved cumbersome processes. Data had to be extracted from one system and pushed into a shared location, often as files through methods like SFTP, or integrated into specific systems like relational databases (RDBMS). These tasks would occur regardless of value or if the end user was even consuming the data. The main priority was ensuring that the data was available at the agreed upon location and time, but if communication between parties failed then data was continued to be shared without value being realized. Furthermore, data governance was frequently deferred to the data consumer, with implicit trust that data usage and security would adhere to the agreed-upon terms.

Delta Sharing has transformed data sharing by allowing organizations to deliver data directly to consumers while minimizing compute costs. Cloud egress fees are only incurred when data is actively consumed, resulting in a cost-efficient data-sharing model. Delta Sharing provides organizations with the capability to finely control access to data and access can be revoked as needed.

Databricks Delta Sharing

In the following section, we will cover how Delta Sharing could be used for both internally and externally sharing of data using Databricks. Delta sharing enables users to directly access data from a provider using the tool and language of choice.

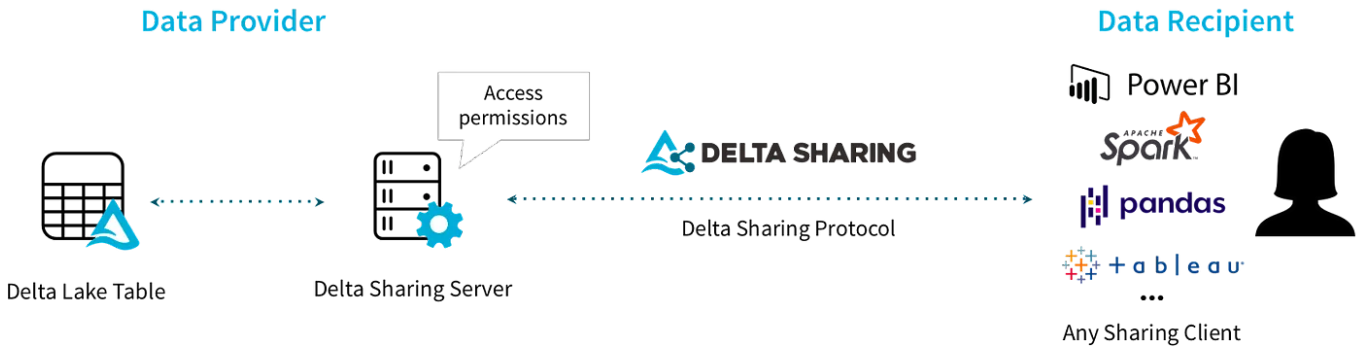


Image Sourced from: https://delta.io/sharing

To highlight the distinct advantages of leveraging Delta Sharing, an open-source data sharing protocol that offers unparalleled flexibility and freedom:

1. **Flexibility and Vendor Neutrality:**
   - Delta Sharing is a fully open-source data sharing protocol that supports batch, incremental, and streaming consumption. One of its standout features is the absence of mandatory vendor-specific compute requirements for data consumption. This means users can employ a diverse array of tools to access and utilize data, effectively eliminating vendor lock-in for data consumption. For example, you can read data from a Databricks customer even if you are not a Databricks customer.
   - Check out this blog by an Oracle Data Platform Specialist demonstrating the data sharing integration between vendors.

2. **Zero Geographic and Vendor Restrictions:**
   - Unlike many vendor-based data sharing solutions, Delta Sharing imposes no geographical or vendor network limitations. Customers are not bound by the requirement that both parties must exist within the same vendor network. Furthermore, Delta Sharing eliminates the need for data replication between regions to facilitate data consumption which requires additional processes and compute to replicate the data. This is particularly advantageous if you need to share data across distant geographic locations, such as from West US to East US, without the

overhead of data replication.
- Please note that users can leverage IP Access Lists in Databricks to enforce security controls on data sources, and could implement something similar for the open source distribution of Delta Sharing.

3. **Zero Geographic Restrictions Apply to Other Clouds:**
- The advantages mentioned above extend not only within a single cloud but also apply when sharing data between clouds and even to on-premises users. With other solutions, data might need to traverse multiple hops and undergo complex processes to be consumed across various clouds, regions, or on-premises locations.

4. **Delta Sharing does not require provider compute:**
- Once data is persisted and made available there are no Databricks compute charges for user consumption. Providers will only need to be concerned about cloud storage fees which are significantly cheaper. This distinction underscores the economic efficiency of the Delta Sharing model.
- Note that when sharing a view via Delta Sharing it may require compute from the data provider in order to materialize the data prior to sharing.

Databricks Data Replication and Sharing Options

This section dives into data sharing solutions tailored for internal use within a single organization, while also shedding light on a recommended best practice for sharing data across clouds. This recommendation particularly shines when there are Databricks deployments spanning multiple cloud environments.

- **Delta Sharing:**
- Delta sharing supports batch, incremental, and streaming workloads.
- Delta sharing abstracts the data location to the users so the end user

may not know the data is being shared for a different location. Delta sharing allows you to read data regardless of which cloud region you are deployed in.
- Delta Sharing integrates seamlessly with the Unity Catalog and supports enterprise governance protocols, enabling central management and auditing of shared data.
- Allows end users to **always** read the most up-to-date version of the data regardless of stream/batch consumption.

- **Table Clones**: Table cloning in Databricks involves replicating a table and its data to another location.
  - Cloning tables allows you to replicate the current version of a table to a secondary location. In addition to sharing data, this can be used for data redundancy, disaster recovery.
  - It's useful for creating point-in-time data snapshots, enabling users to reference data on a set refresh schedule. For example, if data only needs to be refreshed once a day this is a great solution to move to another region/cloud for local users to leverage.
  - Table cloning can occur between clouds and regions.
  - For batch refresh of tables, cloning tables is the most cost-effective way to share data with numerous users in a specific locality where real-time data is not required because clones can be executed in an incremental manner.
  - Table clones do not support in-flight processing and should be used for replication.

- **Spark Read/Write:**
  - Spark Read/Write allows data to be read from one cloud storage account and written to another, regardless of the cloud provider.
  - Similar to Delta Sharing, it supports batch, incremental, and streaming data updates with in-flight transformations.

- Allows users to read data from the cloud they are working in can improve performance and reduce egress fees.
- Native Spark capabilities can retain table history for the target table which allows for rollback but won't necessarily replicate the source table's history.
- It supports in-flight transformations, making it suitable for real-time data processing.

Selecting the right cross-cloud data sharing option depends on your specific use case, budget, and performance requirements. Delta Sharing, Table Clones, and Spark Read/Write each offer unique benefits, allowing organizations to tailor their data sharing strategy to their individual needs. Careful consideration of each use case is essential to make the right choice. As an extremely high-level summary of the topics discussed in this blog, please review the following on when to use which method.

	Delta Sharing	Table Cloning	Spark Read-Write
Batch	Yes	Yes	Yes
Streaming	Yes	No	Yes
Incremental	Yes	Yes	Yes
Requires Data Replication	No	Yes	Yes
In-Flight Transformations	Yes	No	Yes
Requires Compute	No*	Yes	Yes
Multi-Cloud	Yes	Yes	Yes
Cloud Egress Fees	On Read	On Clone	On Write

Capability Overview

If I were a data leader for an organization, this would be my default policy and guidance to my teams:

- Default Policy
  - Avoid excessive data egress from cloud vendors.
  - Data consumers should reference the data in the cloud that they are working.
  - Data is replicated to a second cloud using Table Clones or Spark Read/Write as needed which can be facilitated by Databricks.

- If there are few data consumers for a subset of data or real-time/up-to-date data is needed, then use Delta Sharing.
  - Continuously reading from Delta shares by many users could increase cloud egress costs so it should be on an as needed basis and reads cross-clouds should be minimized.

- If there are many consumers requiring access to real-time/up-to-date data, then use Spark Structured Streaming to stream the data to a table in the appropriate cloud.
  - Streaming the data from one cloud to another allows for real-time data and egress costs are only incurred once, then real-time data can be consumed by many users.
  - Note that since it is possible to stream from a Delta Share as well, this can also be accomplished with Delta Sharing on the consumer side. Specifically, with Delta Sharing the consumer is responsible for the read/write while traditional methods the publisher is required.

Ultimately, the success of cross-cloud data sharing relies on a combination of technological solutions, organizational policies, and best practices. It's important to strike a balance between flexibility, performance, cost-effectiveness, and compliance to ensure that your chosen data sharing method aligns with your organization's unique requirements and goals.

Disclaimer: these are my own thoughts and opinions and not a reflection of my employer

Delta Lake | Delta Sharing | Databricks | Data Sharing

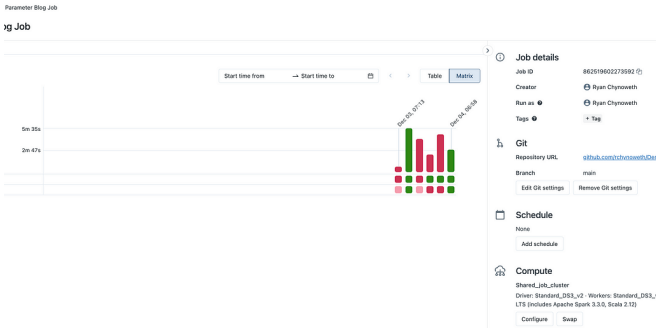
Written by Ryan Chynoweth

Edit profile

312 Followers

Senior Solutions Architect Databricks — anything shared is my own thoughts and opinions

More from Ryan Chynoweth



Ryan Chynoweth

Delta Sharing: An Implementation Guide for Multi-Cloud Architecture

Introduction

8 min read · 3 days ago

3 2

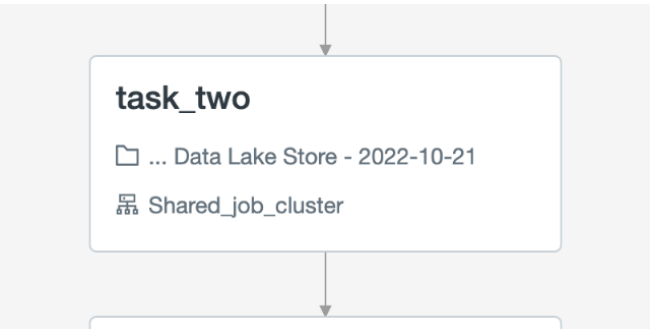
Ryan Chynoweth

Task Parameters and Values in Databricks Workflows

Databricks provides a set of powerful and dynamic orchestration capabilities that are...

11 min read · Dec 7, 2022

50 3



Ryan Chynoweth

Converting Stored Procedures to Databricks

Special thanks to co-author Kyle Hale, Sr. Specialist Solutions Architect at Databricks.

14 min read · Dec 29, 2022

116 5

Ryan Chynoweth

Recursive CTE on Databricks

Introduction

3 min read · Apr 20, 2022

33

See all from Ryan Chynoweth

Recommended from Medium

Daan Rademaker

Do-it-yourself, building your own Databricks Docker Container

In my previous LinkedIn article, I aimed to persuade you of the numerous advantages o...

7 min read · Oct 16, 2023

26

Nnaemezue Obi-Eyisi

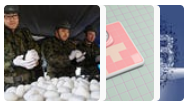
Unveiling the Secrets: External Tables vs. External Volumes in...

While reviewing the Databricks documentation about Unity Catalog, I came...

✦ · 7 min read · Sep 25, 2023

39 1

Lists



Staff Picks

547 stories · 597 saves

Stories to Help You Level-Up at Work

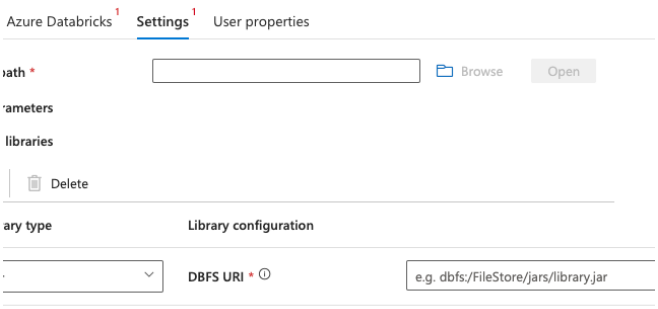
19 stories · 395 saves

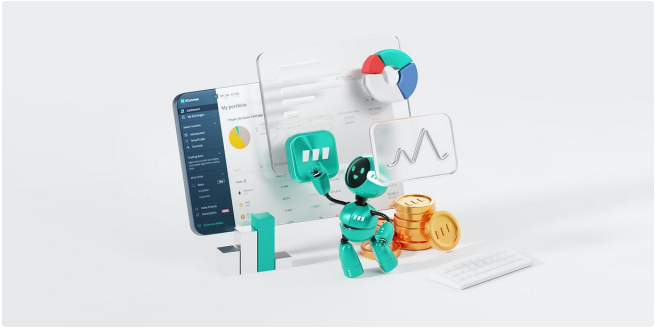
Self-Improvement 101

20 stories · 1146 saves

Productivity 101

20 stories · 1047 saves





Matt Bradley

Azure Data Factory tips for running Databricks jobs

Following on from an earlier blog around using spot instances with Azure Data...

4 min read · Nov 2, 2023

14

Arpine K in Open Data Discovery

Data Quality Dashboard

Informed decision making relies on data: a crucial asset for business, however, not all...

4 min read · Dec 21, 2023

9

Manasreddy

How to pass: Databricks Data Engineer Professional Certification

Conquering the Databricks Data Engineer Professional Exam: A Definitive Guide

3 min read · Aug 24, 2023

8

Shane Hender in Zendesk Engineering

Moving from DynamoDB to tiered storage with MySQL+S3

Originally we implemented a feature to persist an event-stream into DynamoDB to allow...

8 min read · Nov 8, 2023

187 7

See more recommendations