Qlik

| WHITE PAPER

Best Practices Guide

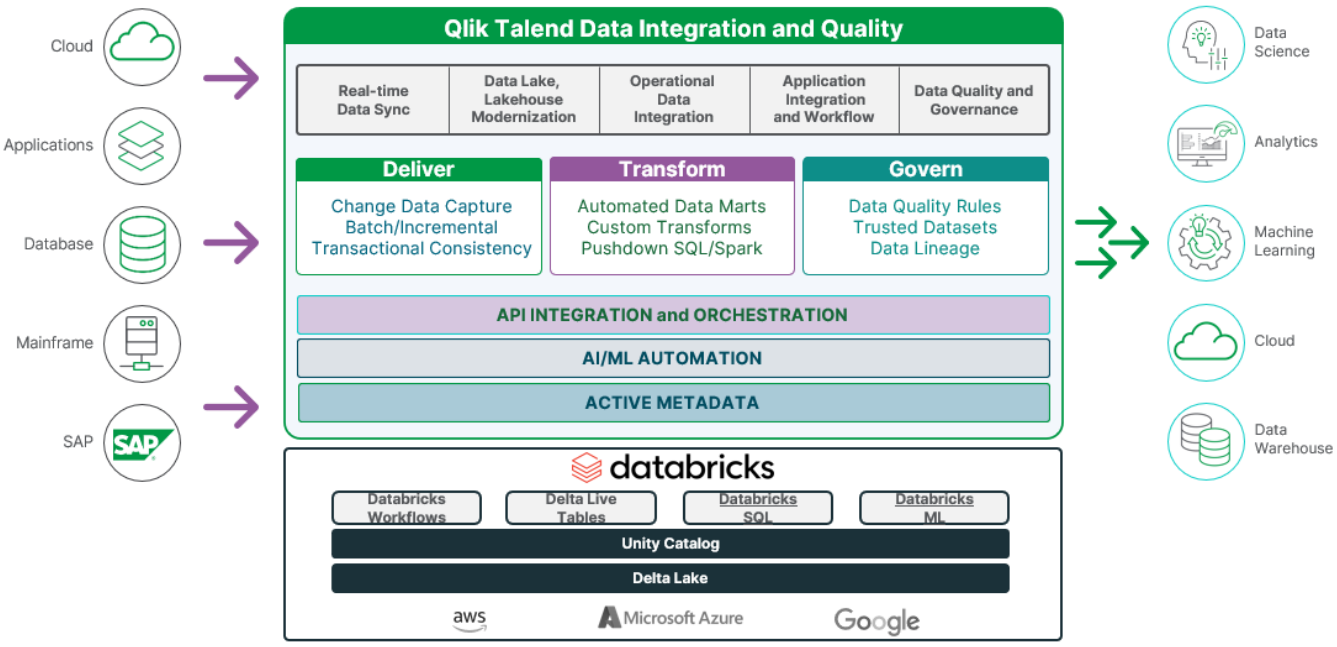
Qlik Talend™ Data Integration and Databricks®

Table of Contents

Introduction

Qlik Talend Data Integration solutions accelerate machine learning (ML), artificial intelligence (AI), and DataOps initiatives with Change Data Capture (CDC), and transformation technology that ensures continuous data streams from multiple data sources to the Databricks Lakehouse Platform ready for AI and Analytics consumption.

Qlik Talend Data Integration Solutions



Qlik Replicate and Qlik Talend Cloud Data Integration are two solutions from Qlik that enable enterprises to manage their data across different sources and platforms. Qlik Replicate is a data integration software that allows users to replicate and update data in real time from on-premises and cloud sources to cloud data warehouses without manual coding or scripting. Qlik Cloud Data Integration is a cloud-based service that provides the ability to create data pipelines to perform various data integration tasks, such as landing, registering, transforming, and unifying data. Both solutions support a wide range of data sources and destinations, such as relational databases, big data platforms, SAP, Mainframes, cloud storage, and SaaS applications. However, there are some differences between them in terms of features, pricing, and deployment options.

Some of the main differences are:

- Qlik Replicate is a standalone product that can be installed on-premises or on the cloud, while Qlik Talend Cloud Data Integration is a fully managed service that runs on the Qlik Cloud platform.
- Qlik Replicate offers more advanced features for data replication, such as change data capture (CDC), delayed merges, schema evolution, and conflict resolution. While Qlik Talend Cloud Data Integration focuses more on data transformation and unification, such as data cleansing, enrichment, blending, and profiling.

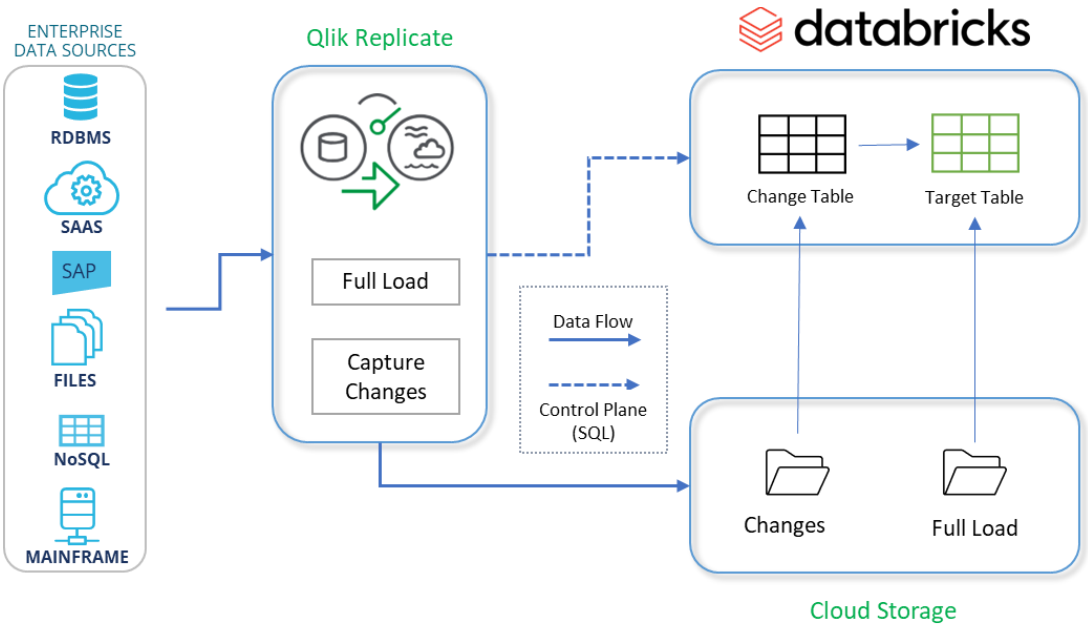
In summary, Qlik Replicate and Qlik Talend Cloud Data Integration are complementary solutions that can be used together or separately depending on the use case and the data architecture of the enterprise. Both solutions aim to provide fast, reliable, and scalable data integration capabilities for modern AI and Analytics needs.

Qlik Replicate

Qlik Replicate® continuously automates CDC data movement from multiple data sources (e.g., Oracle, Microsoft SQL Server, SAP, Mainframe, and more) to the Databricks Lakehouse Platform. It helps the customer avoid the heavy lifting associated with manually extracting data, transferring it via API/script, and then slicing, staging, and importing it.

In this architecture, Qlik Replicate performs the following functions:

1. Instantiate the Target
    a. Create target tables in DELTA format with proper data types translated from the source
    b. Perform an initial/full load from the source and send the data into the storage layer
    c. Send Spark SQL to Databricks to load the data from the storage layer and convert it into tables using the delta format
2. Capture and Apply Changes
    a. Capture changes using log-based CDC from the source
    b. Deliver and APPLY changes (Insert / Update / Deletes) to the target DELTA tables (using the storage layer as intermediate staging)



Qlik Cloud Data Integration

Qlik Cloud Data Integration is an iPaaS (Integration Platform as a Service) offering that provides the ability to create data project pipelines to perform a variety of data integration tasks in support of your data architecture and AI, and Analytics requirements.

- Data pipelines - You can leverage real-time, log-based change data capture with secure connection to on-premises data sources behind firewalls or use full load capabilities for SaaS data sources. Once you have onboarded data, you can apply transformations for fit-for-purpose output or automate patterns like data mart facts and dimensions. External views and live views are generated for data consumption. Qlik Cloud Data Integration also generates a full type 2 historical data store (HDS).
- Data Replication Tasks - Replicate data from any compatible source to any supported destination. Data can be transformed and remain consistently updated using Change Data Capture (CDC) techniques. There is the capability as well to deliver the data into a data lake delivering data into Amazon S3, Azure Data Lake Storage, or Google Cloud Storage

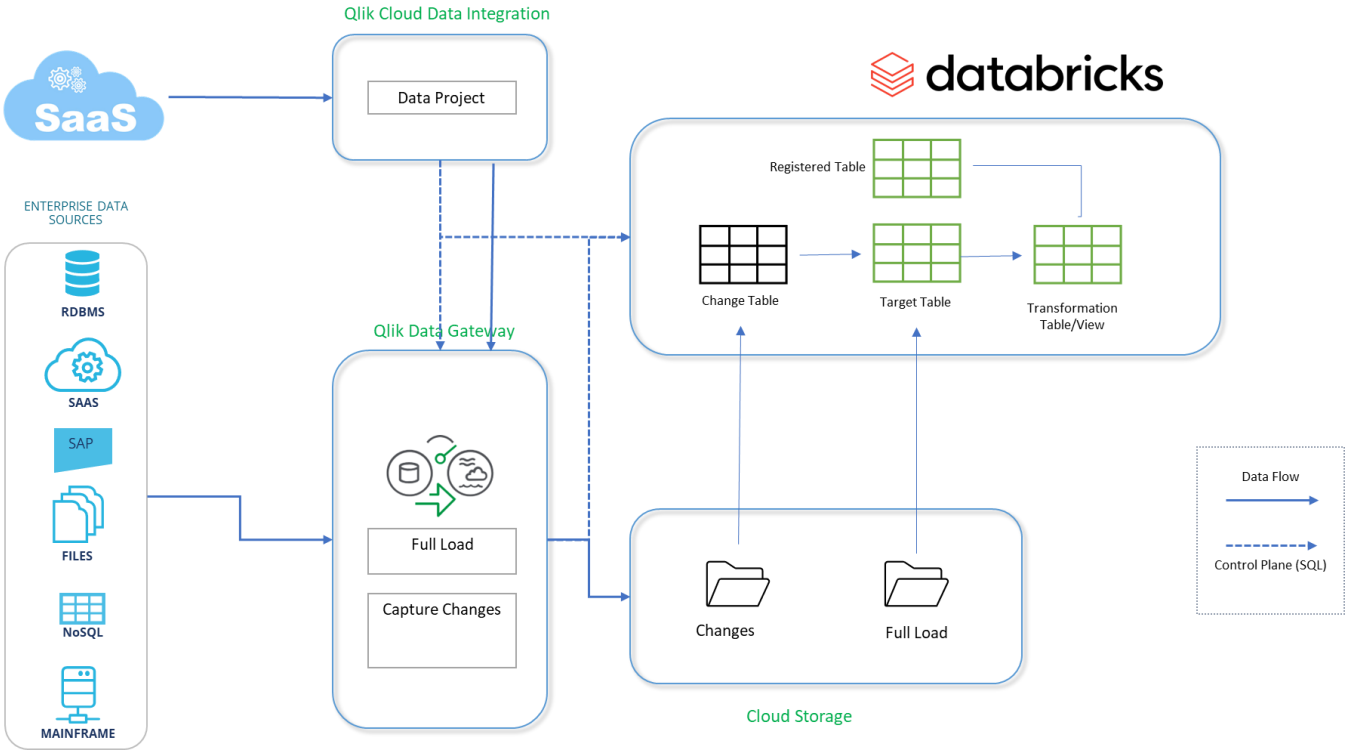
The refined data output from Qlik Cloud Data Integration can be used for many purposes:

- Real-time movement from all enterprise sources including relational databases, SAP, Mainframe, and SaaS applications.
- Data transformation using ELT (Extract/Load/Transform) using a no-code approach without the need for additional 3rd party solutions.

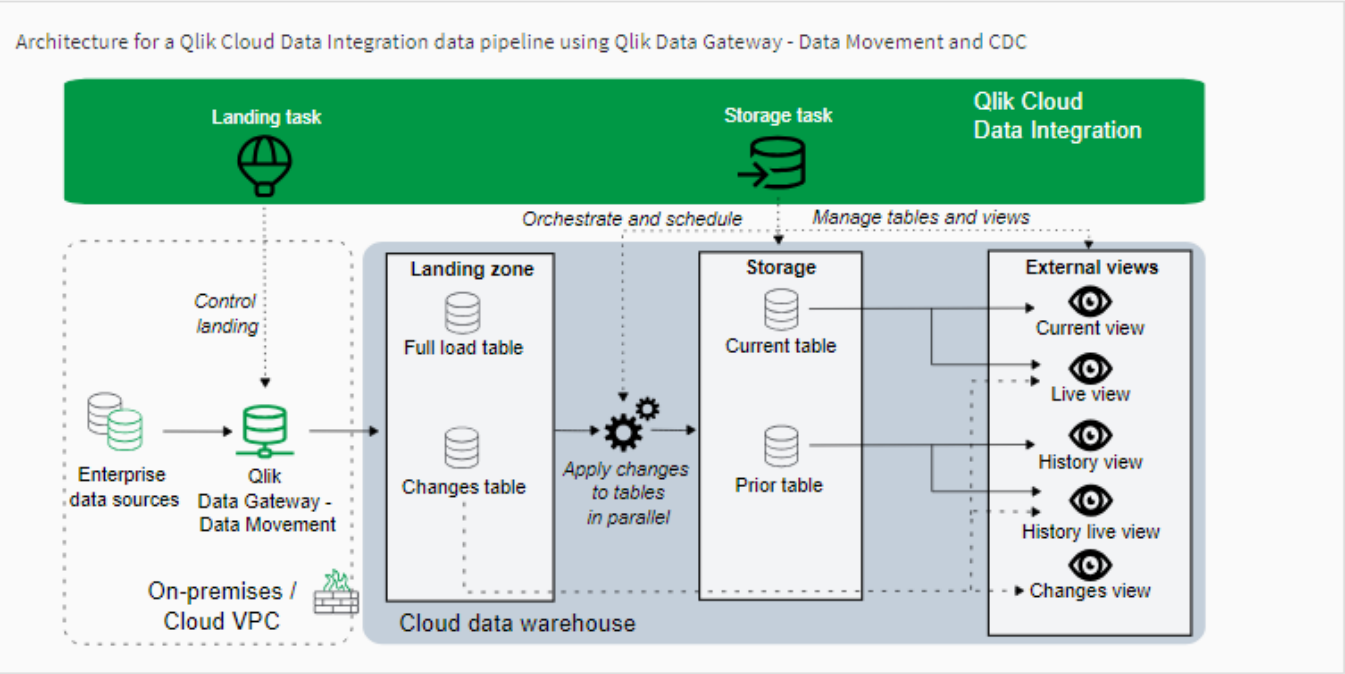
- Automated creation of datamarts for analytics in the Databricks Lakehouse.
- Modernization of your data repository to support AI, Machine Learning and other initiatives.

In this architecture, Qlik Cloud Data Integration performs the following functions:

- Instantiate the Target.
    - Create target tables in DELTA format with proper data types translated from the source tables
    - Perform an initial/full load from the source sending the data to the storage layer
        - SaaS applications – directly
        - Relational Databases – using Data Gateway
    - Send Spark SQL to Databricks to load the data from the storage layer and convert it into tables using the delta format
- Capture and Apply Changes.
    - Capture changes using log-based CDC from the source
        - SaaS applications – directly
        - Relational Databases – using Qlik Data Gateway
    - Deliver and APPLY changes (Insert / Update / Deletes) to the target DELTA tables (using the storage layer as intermediate staging)
- Execute the Transformations Sending Sparksql to Databricks



Another way of seeing this architecture is through the concept of tasks with specialized functions



1. Landing Tasks - oversees the seamless transfer of data from various sources to the designated landing zone. The illustration diagram illustrates the utilization of Qlik Data Gateway - Data Movement for accessing data sources via Change Data Capture (CDC) to ensure the data remains current. Additionally, Qlik Cloud Data Integration source connections can be employed to execute full loads, allowing for scheduled periodic reloads.
2. Storage Tasks - oversees the application of data to storage tables, including the creation and administration of both tables and external views. This crucial task plays a pivotal role in maintaining data integrity and accessibility within the Qlik Cloud Data Integration environment. The storage task not only governs the timing of data application but also ensures the seamless integration of information into the storage infrastructure, enhancing the overall efficiency and functionality of the Qlik Cloud Data Integration platform.
3. Transformation Tasks - Within your data pipeline, you can generate data transformations that are both reusable and rule-based. These transformations can be seamlessly incorporated into your data onboarding process or set up as reusable transformation data tasks. The flexibility extends to performing row-level transformations and crafting datasets using custom SQL, which can either materialize as tables or manifest as dynamic views applying transformations on the fly.
4. Data Mart Tasks - After successfully onboarding the data, it's possible to generate data marts utilizing the information sourced from either the Storage or Transform tasks. Tailoring to the business requirements, multiple data marts can be created. Ideally, these data marts should serve as repositories for aggregated data, gathered for analytical purposes within a specific department or unit

of an organization, such as the Sales department or even exposed as features to be consumed by ML processes.

Implementation guidance

As mentioned previously in this document, Qlik Replicate and Qlik Cloud Data Integration are two solutions that can be used either together or separately. The decision to use them in combination or individually depends on the data architecture and needs of the enterprise. It is important to consider customer requirements and use cases to determine the most efficient and effective architecture.

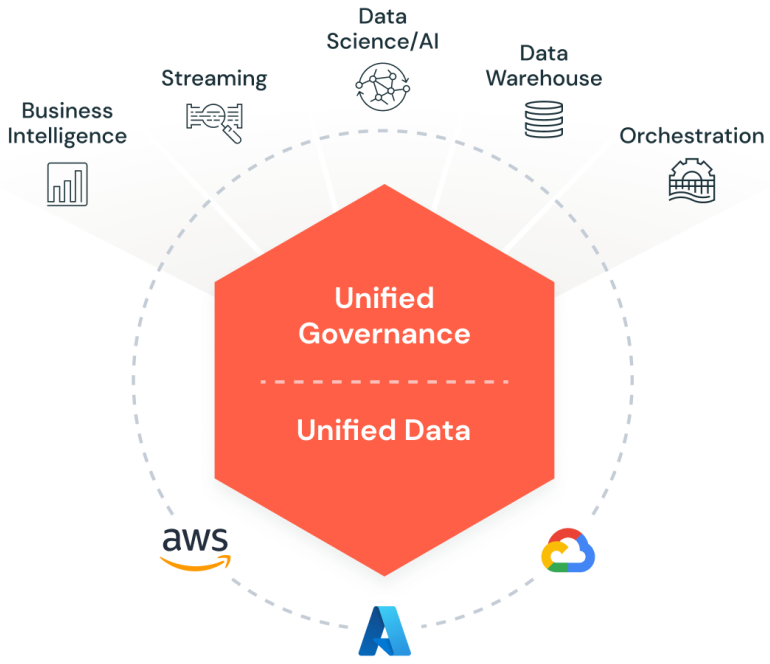
- One to many topologies require currently the usage of Qlik Replicate
- SaaS sources are supported only by Qlik Cloud Data Integration

There may be cases where both Replicate and Qlik Cloud Data Integration can be utilized together. For instance, Replicate can be used to feed data into Databricks from a source that is not currently supported by Qlik Cloud Data Integration. The registered data can then be used as inputs for data pipelines created using Qlik Cloud Data Integration.

Databricks Data Intelligence Platform

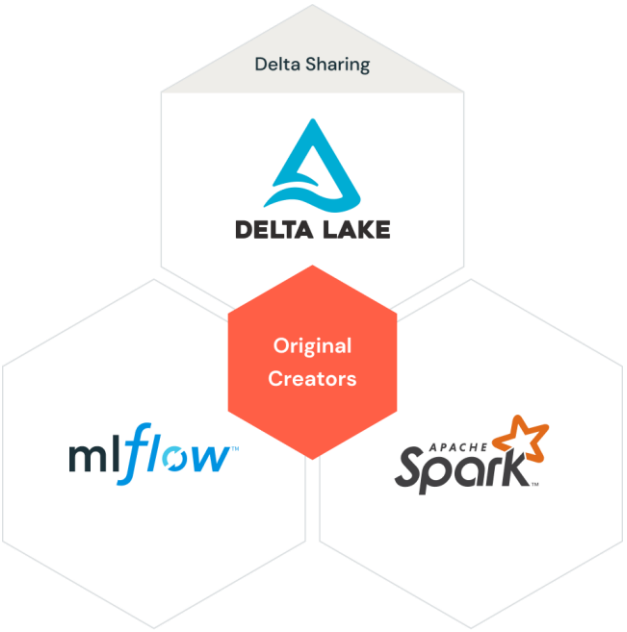
The foundation of the Databricks Data Intelligence Platform lies in the lakehouse architecture, a revolutionary blend of data lakes and data warehouses. This innovative approach is geared towards minimizing costs and expediting the realization of data and AI goals.

Embracing open-source principles and adhering to open standards, the lakehouse architecture streamlines data infrastructure by removing the historical barriers that often complicate the realms of data and AI. By doing so, it offers a more cohesive and efficient environment for managing and leveraging your data resources.



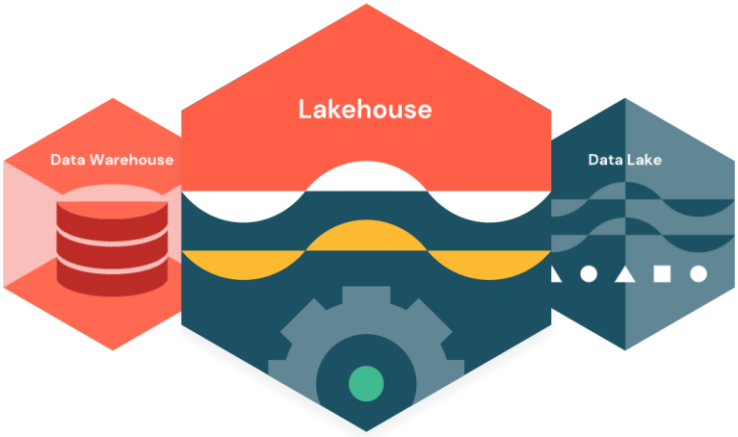
Unified

A unified architecture encompassing integration, storage, processing, governance, sharing, analytics, and AI. A singular methodology for handling both structured and unstructured data. A comprehensive perspective on data lineage and provenance from start to finish. A cohesive toolkit accommodating Python and SQL, notebooks and IDEs, batch and streaming processes, across all major cloud providers.



Open

Within the Databricks framework, control over data is consistently maintained, ensuring independence from proprietary formats and closed ecosystems.



The foundation of the lakehouse architecture relies on widely embraced open-source projects such as Apache Spark™, Delta Lake, and MLflow. It enjoys global support through the Databricks Partner Network. Additionally, the Delta Sharing feature presents an open solution for securely sharing real-time data from the lakehouse to any computing platform. This is achieved without the need for data replication or intricate extract, Transform, Load (ETL) processes.

Scalable

The automatic optimization for performance and storage is meticulously designed to ensure the lowest Total Cost of Ownership (TCO) among data platforms, concurrently achieving world-record-setting performance for data warehousing and Artificial Intelligence (AI) use cases. This extends to the application of generative techniques such as Large Language Models (LLMs).

Irrespective of organizational scale, Databricks is engineered to effectively address the operational requirements of businesses, ranging from startups to global enterprises.

SQL Warehouses x General Compute Clusters

Qlik solutions supports both Databricks SQL Warehouses and Compute Clusters. They are two different ways of processing data in the cloud. In both cases Qlik solutions will send SparkSQL commands to process the data, not relying on other supported features (like notebooks in Scala for example). The choice between Databricks SQL warehouses and general compute clusters depends on the specific requirements and goals of each project. Some factors to consider are:

Data volume
CDC frequency: SQL Warehouses are faster to spin up when actioned the first time, and clusters can be slower to answer the very first command.
Scalability and elasticity: SQL warehouses can automatically scale up or down to meet the demand of concurrent users and queries. General compute clusters can also scale but require more manual intervention and tuning.
Security and governance: SQL warehouses provide built-in security features such as encryption, authentication, authorization, auditing, and compliance. General compute clusters can also implement security measures but require more configuration and management.

In summary, Databricks SQL warehouses and general compute clusters are both powerful and reliable solutions for data processing in the cloud. However, they have different strengths and weaknesses that should be carefully evaluated before choosing one over the other.

Recommendations for General Compute Clusters for Qlik Solutions

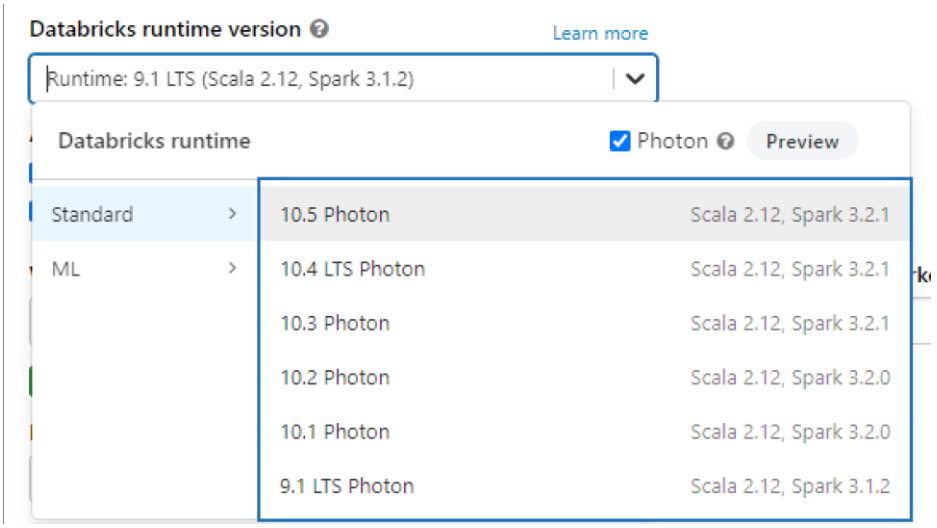
NOTE
The recommendations below are for reference and are based on projects and POCs conducted by Qlik and its partners concerning specific requirements. Several factors like network topology, latency, table structure, update frequency, driver versions, etc. may affect the necessary settings for your particular use case. Customers are advised to perform the necessary scoping and diligence to determine their configurations.

1. Databricks Runtime

Always check Qlik Replicate and Qlik Cloud Data Integration (http://help.qlik.com) to see which Databricks Runtime is supported when you are configuring your cluster.

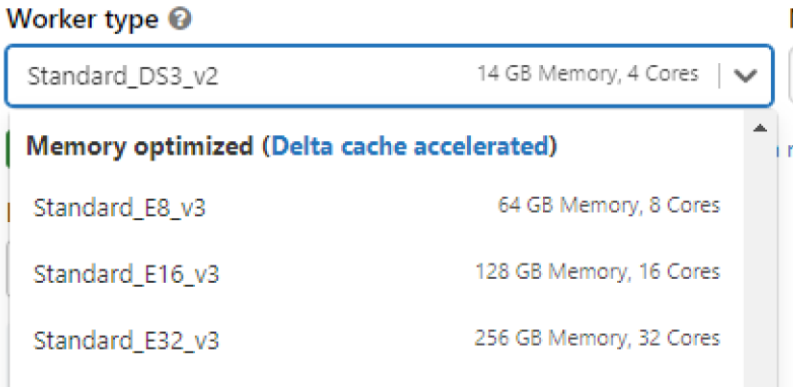
2. Databricks Runtime Version Supporting Photon

When you are configuring your cluster select “Photon” for the Databricks runtime version that will support your general-purpose cluster. Photon is the native vectorized query engine on Databricks, written to be directly compatible with Apache Spark. Photon is part of a high-performance runtime that runs your existing SQL and DataFrame API calls faster and reduces your total cost per workload. For a further discussion about Photon, please refer to this document https://docs.databricks.com/runtime/photon.html

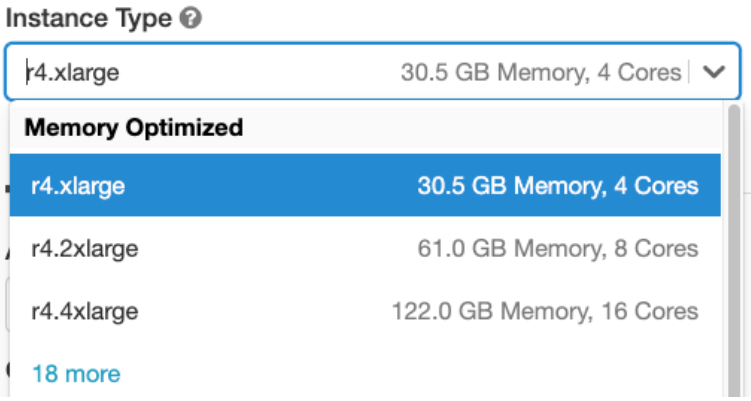


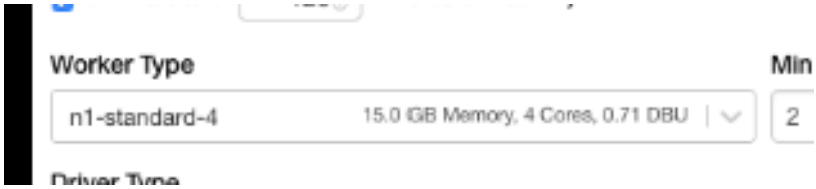
3. Select “Memory optimized – Delta cache accelerated”

   When you are configuring your cluster, make sure you select the “Memory optimized – Delta cache accelerated” Worker type.



(*) the list above is based on Azure Databricks, this may change if you´re using AWS or GCP



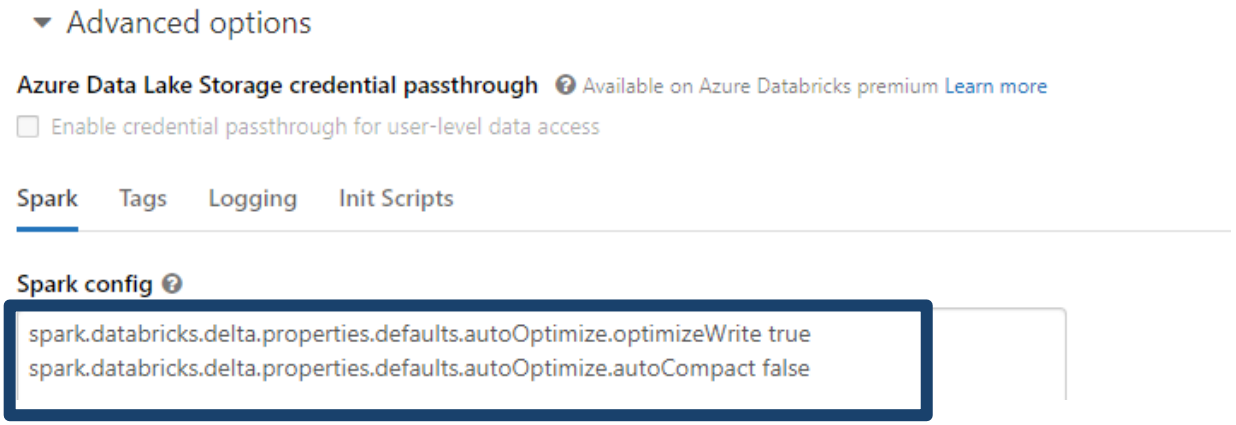


4. Configure Auto-Optimize Options

Add a configuration to your cluster to enable optimizeWrite and disable autoCompact. Disabling autoCompact is necessary to prevent serial compaction from being triggered by real-time CDC updates (which can lead to increased latency). To do that, add the lines below to your Spark section of Advanced options of your cluster.

```
spark. data bricks.delta.properties.defaults. auto-optimize.optimizeWrite true
spark.databricks.delta.properties.defaults.autoOptimize.autoCompact false
```

Please check https://docs.databricks.com/clusters/configure.html for more information about configuring your cluster.



5. Optimize Tables Regularly

   It is important to schedule a notebook to OPTIMIZE tables in your Delta Lake. This will improve query speed for the data landed. Please consult this documentation: https://docs.microsoft.com/en-us/azure/databricks/delta/optimizations/file-mgmt for samples of notebooks to optimize the tables.

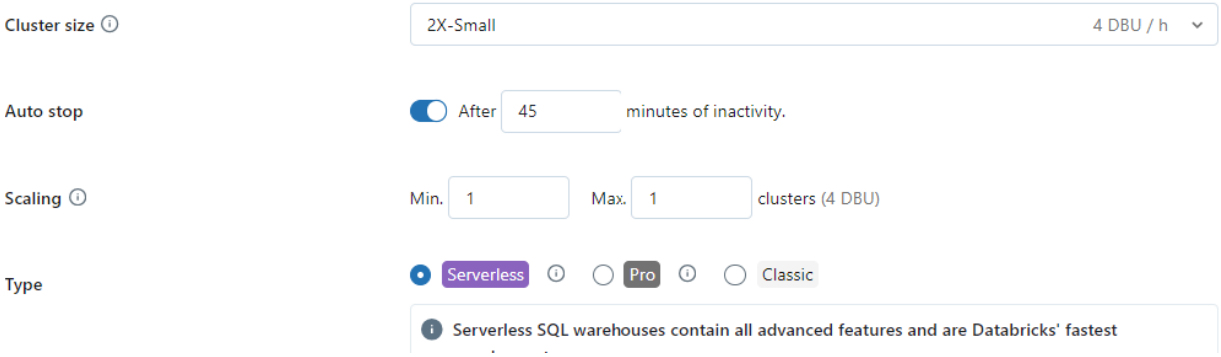
6. Autoscaling

Due to the variable workload volumes that the CDC presents, the recommendation is to review your configuration based on the workload and testing with your tasks, monitoring and then increasing or decreasing based on the usage. Please refer to Databricks documentation (https://docs.databricks.com/clusters/clusters-manage.html#monitor-performance) for how to monitor cluster performance.

Recommendations for SQL Warehouses for Qlik Solutions

NOTE
The recommendations below are for reference and are based on projects and POCs conducted by Qlik and its partners concerning specific requirements. Several factors like network topology, latency, table structure, update frequency, driver versions, etc. may affect the necessary settings for your particular use case. Customers are advised to perform the necessary scoping and diligence to determine their configurations.

SQL Warehouses have much fewer options to configure at the warehouse level (compared to clusters). Available configurations.



1. Warehouse Type

As of when this document was written there are three warehouse types. Please refer to this document What are SQL Warehouses? for a general discussion about them

	Photon Engine	Predictive IO	Intelligent Workload Management
Serverless	X	X	X
Pro	X	X	
Classic	X		

From a performance and concurrency perspective, the general recommendation is to use a Serverless warehouse to increase the general performance of your task. Some environments and accounts don´t have this option, in this situation, the recommended is a Pro warehouse.

2. Scaling

   Adjust this parameter to increase based on tasks parameter "Maximum number of tables to load in parallel" (Replicate) or "Maximum number of database connections" (Qlik Cloud Data Integration). The general rule of thumb is to have one warehouse cluster to process 2 to 3 tables or connections in parallel.

3. Cluster Size

   This parameter is highly dependent on the data being processed. Several parameters may impact the general performance like several tables, the number and column types of each table, update frequency, etc.

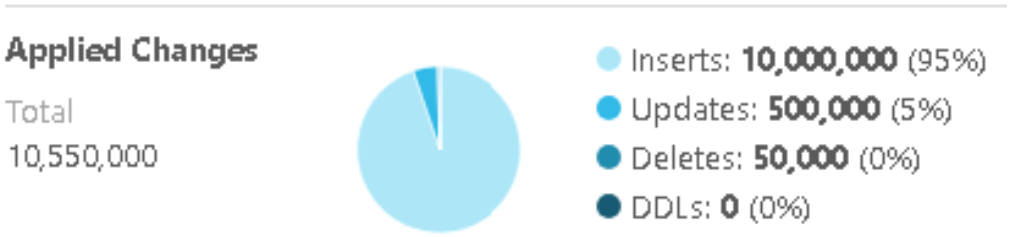
   The general recommendation is to start with a size that has a good expected cost x performance (like a medium for example) and conduct some testing adjusting this parameter (up or down) comparing with the baseline.

Performance comparison between SQL Warehouses x General Compute Clusters

NOTE
This test was conducted in a laboratory environment and does not represent any real live environment. Results may vary depending on the type of sources, topology, volumes, record size, source database tunning, and other variables.

Environment:

- Source
  - PostgreSQL running on a VM
  - 1 Table with 7 columns and 36 million records
  - Primary Key(1 columns)
  - CDC script testing with the profile below



- Targets (Databricks on Azure) – Staging on ADLS gen2

SQL Warehouse		Cluster
Type	Pro	Summary
Cluster size	Medium	1-4 Workers: 64-256 GB Memory, 8-32 Cores
Auto stop	After *120* minutes of inactivity	1 Driver: 14 GB Memory, 4 Cores
Scaling	Cluster count: Active *2* Min *1* Max *4*	Runtime: 10.4.x-scala2.12
Channel	Current (v 2023.40)	Photon, Standard_E8d_v4, Standard_DS3_v2, 5–18 DBU/h
Spot instance policy	Cost optimized	

- Qlik Replicate November 2023 (2023.11.0.149) on Windows
- Process
  - Two tasks (same source and different target endpoints)
  - Started the full load on task 1 and waited for completion.
  - Started the full load on task 2 and waited for completion.
  - Started the SQL script simulating the changes. Both tasks run in parallel and deliver to different schemas on Databricks

Results:
Using Qlik Enterprise Manager to collect all the statistics, we observed that there is no significant difference in performance between using SQL Warehouses and Clusters. From a cost perspective, SQL Warehouses are usually more cost-effective, which can indicate their usage over clusters.

Task Name	Server Name	Server Utilization			Full Load		Change Processing		
		Avg Disk Usa...	Avg Memory	Avg Task CP...	Max Load D...	Avg Target T...	Total Applied...	Avg Apply T...	Avg Apply L...
BP Cluster	CLEAS Repli...	50.53 MB	590.48 MB	0.7	00:15:52.509	88,989	30,545,514	92,558	10
BP Warehouse	CLEAS Repli...	66.56 MB	700.27 MB	0.7	00:16:12.018	88,999	28,640,297	108,994	20

Recommendations for Qlik Cloud Data Integration

1. Table Selections in A Task

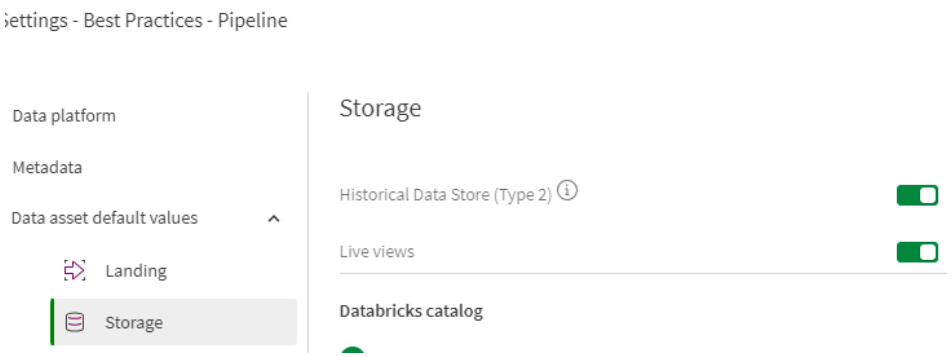
   Databricks recommends isolating large or wide (many columns) tables that do a lot of processing for their tasks. Using this approach makes it easier to monitor the performance or allocate a warehouse to a specific task.

2. Transformations In a Task

   If your objective is to optimize the throughput of highly transactional data being ingested into the lakehouse, it is recommended to minimize the transformations at the task level. This approach allows you to land the data as it is into the lakehouse and then leverage all the available data engineering capabilities to perform the transformations. This process is technically referred to as converting an ETL (Extract-Transform-Load) into an ELT (Extract-Load-Transform).

3. Historical Data Store (Type 2) at Storage / Transform

   Qlik Cloud Data Integration will by default create Historical Data Store (Type 2) assets that retain and manage current and historical data over time, based on the Type 2 SCD (Slowly Changing Dimension) concept. All versions of a record are retained, including deletions, with dates indicating the period each record was active. If this information is not relevant, you can disable this feature in the Storage or Transform section of the project settings. This will save resources (space and computing) from your workspace.



4. Materialization on Transformation Tasks

   Data generated from a transformation task can be exposed in two ways:

   - Views - where all the queries against this entity will be executed against the original tables from the storage zone

- Tables – where the tables will be populated in a scheduled way based on the data from the storage zone.

The choice between them will be based on how frequently the transformed data will be accessed. Less frequently transformed data will be likely exposed as views, more frequently transformed data should be persisted as tables. If there is a need to use both, there is the option to have two (or more) transformation tasks, one based on views and the other(s) based on tables.

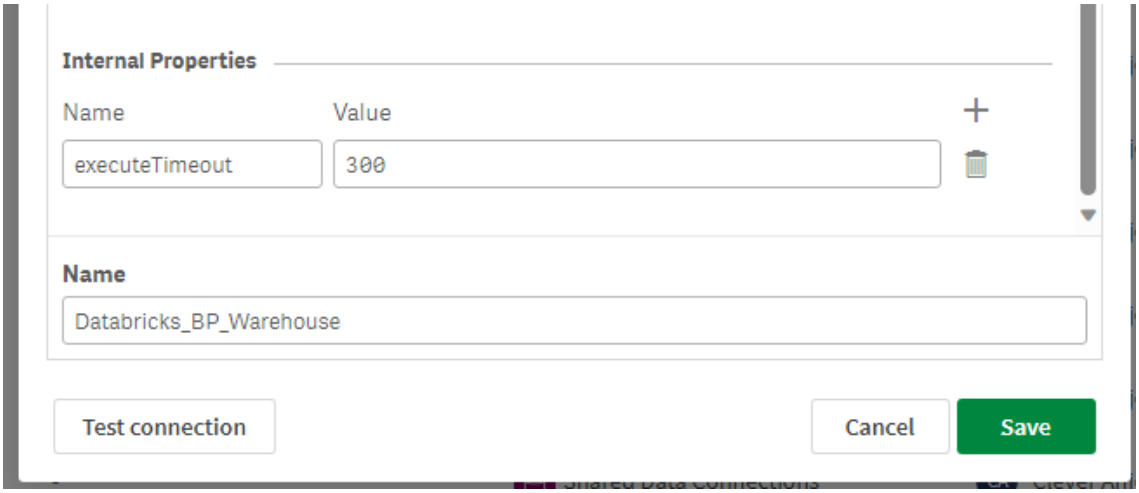
5. Using Live Views

Live views incorporate data from change tables that have not yet been applied to the current or prior tables. This feature allows users to access data with reduced latency without the need for frequent application of changes. Delaying the merge operation also leads to cost savings and decreased processing demands on the target platform.

Additionally, live views offer the advantage of not requiring the compute tier to be always operational. Latency can be enhanced as there is no longer a need to apply changes throughout the day. Newly inserted records become immediately available in the live views once they are accessible in the changes table and the storage task might run less frequently saving clusters/warehouse resources.

6. Timeout At the Connection Level

To ensure the optimal operation of Qlik Cloud Data Integration, it is crucial to configure an internal property named executeTimeout with a value greater than 300. This configuration ensures that the Qlik Cloud Data Integration system will maintain a waiting period of at least 5 minutes before registering a failure. Consequently, this provides sufficient time for the warehouse to initialize if it was previously in a halted state. This is particularly important in scenarios where the warehouse requires a longer startup time.



Recommendations for Qlik Replicate

1. Table Selections in A Task

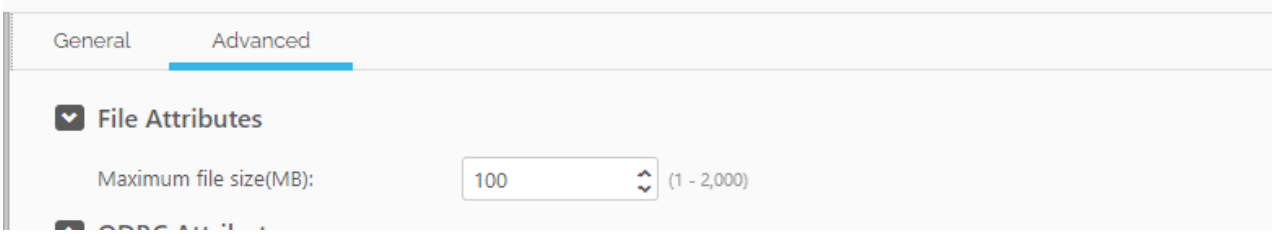
   Databricks recommends isolating large or wide (many columns) tables that do a lot of processing for their tasks. Using this approach makes it easier to monitor the performance or allocate a cluster to a specific task.

2. Transformations In a Task

   If your objective is to optimize the throughput of highly transactional data being ingested into the Lakehouse, it is recommended to minimize the transformations at the task level. This approach allows you to land the data as it is into the Lakehouse and then leverage all the available data engineering capabilities to perform the transformations. This process is technically referred to as converting an ETL (Extract-Transform-Load) into an ELT (Extract-Load-Transform).

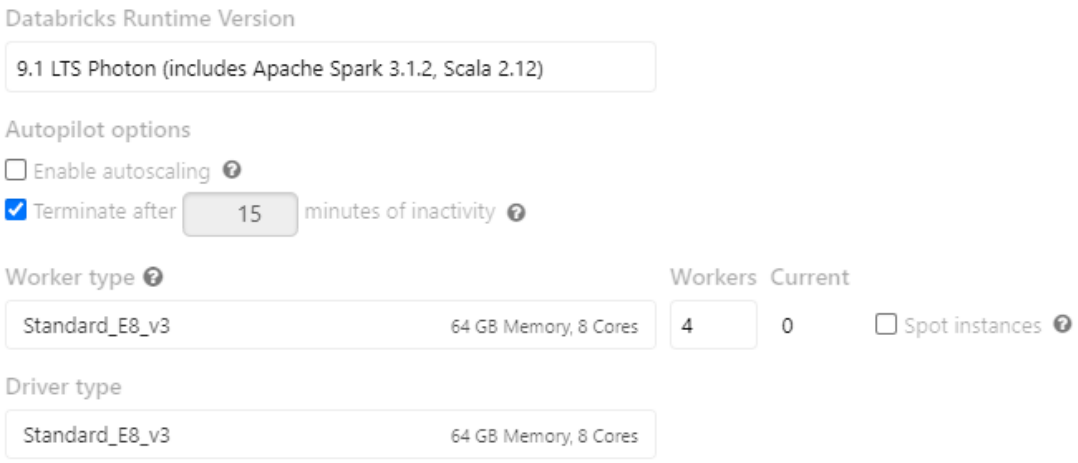
3. File Size Configuration

   There is a Qlik Replicate parameter at the connection level that could increase the data throughput. It is called Maximum file size(MB) and it is located under the Advanced settings of your connection

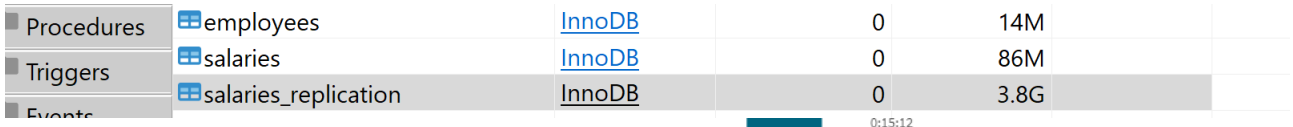


   The default value is 100Mb and this parameter indicates the file size that is uploaded to the staging area before being loaded into a table. You can see below the impact of changing this parameter for a table with 100M records (approx. 3.8 GB data on the source). There is no “golden rule” for this parameter, but usually, a bigger file size increases the performance of the data transfer which is very important during the initial full load.

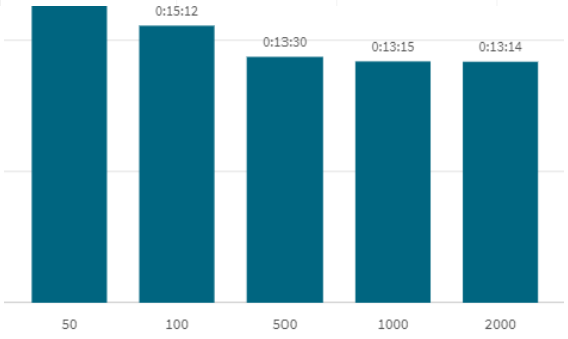
Cluster Configuration:



Source Table (Azure RDS Mysql)



Maximum file size (Mb)	Elapsed Time Full Load
50	00:19:03
100	00:15:12
500	00:13:30
1000	00:13:15
2000	00:13:14



As shown above, there was a very good improvement when increasing the file size from the default value (100MB) to 500MB, though additional increases beyond 500MB in this test had much less impact on performance.

4. Batch Tuning Settings

Qlik Replicate micro-batch changes for optimized delivery to Databricks Delta and batch tuning configuration for a task impacts the size of the micro-batch sent to Databricks.

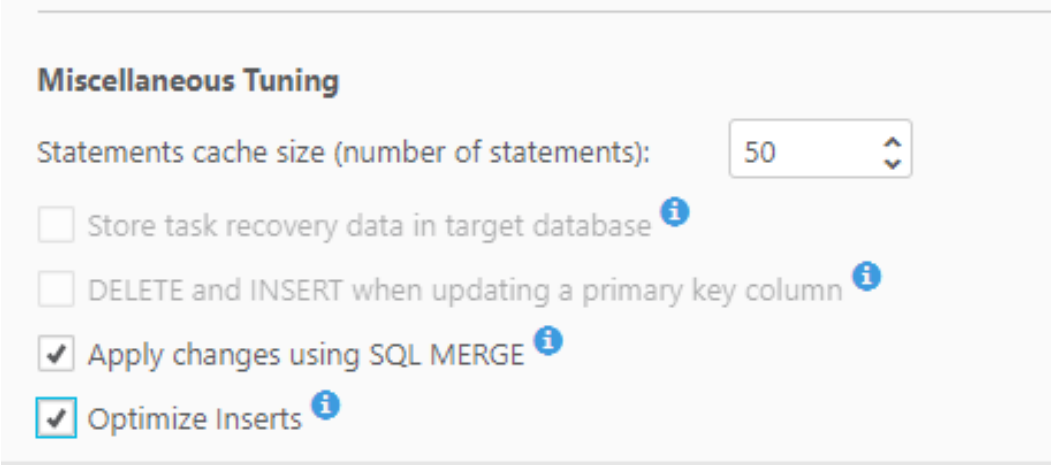
a. Change Processing Mode: Only Batch Optimized Apply is supported for Databricks targets.
b. Apply batched changes to multiple tables concurrently: This option configures the number of threads that will work in parallel to upload and apply data to Databricks. The default value is 5, with a maximum of 50. Increasing this value can improve your throughput when there are many tables with CDC in a given batch however it may require additional cluster resources. Please review the limitations of this mode in the Replicates help guide.
c. Apply batched changes in interval settings: Configure the time and size of the micro-batch.

d. Longer than (seconds): This specifies the minimum amount of time to wait between each application of batch changes. The default value is 1 and typically is too low a value for Databricks delta apply processes. Increasing this value decreases the frequency with which changes are applied to the target while increasing the size of the batches, essentially creating larger batches at the expense of some additional latency. It is recommended to start with a value of 60 and increase even further if some additional latency is acceptable. In some cases waiting for larger batches can improve throughput and latency

e. But less than (seconds): This value specifies the maximum amount of time to wait between each application of batch changes (before declaring a timeout). In other words, the maximum acceptable latency. The default value is 30. This value determines the maximum amount of time to wait before applying the changes after the Longer than (seconds) value has been reached. It is recommended to configure this value to 120 (combined with a Longer Than value of 60 and tune the value even higher if more latency is acceptable.

f. Force apply a batch when processing memory exceeds (MB): this setting specifies the maximum amount of memory to use for pre-processing in Batch optimized apply mode. The default value is 500. For maximum batch size, set this value to the highest amount of memory you can allocate to Qlik Replicate. It is recommended to start with a value of 2000 and consider tuning higher if there are enough resources on the Qlik Replicate server.

g. Apply Changes using Merge: this enables the task to use SQL MERGE commands to speed the ingestion into the Lakehouse.

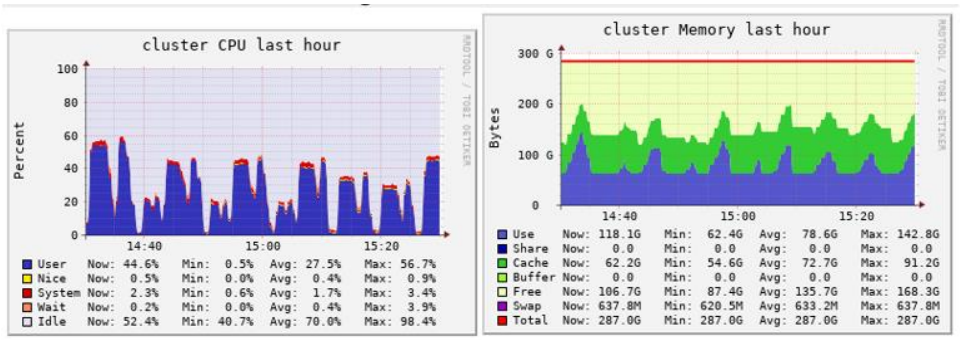


5. Partition Large Tables

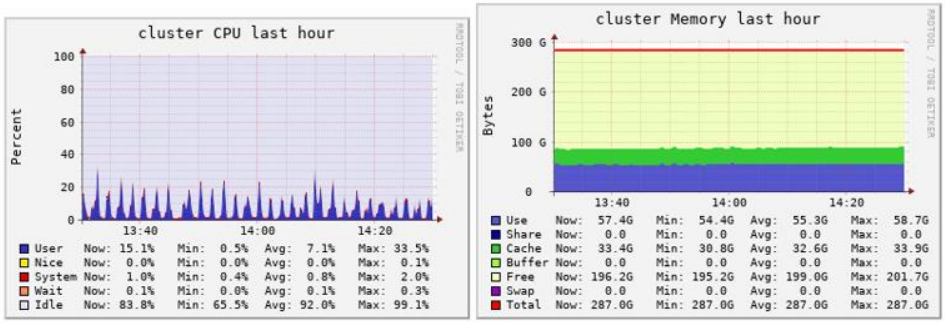
Databricks provides the ability to partition Delta tables. It is recommended to partition large tables that could be a bottleneck in the application process. Qlik Replicate does not currently support configuring target partitioning within the task. The target Delta table should be created by Qlik Replicate and then re-created with the appropriate partition columns. If a table is defined as partitioned, it is recommended to set the task to perform a TRUNCATE for full loads.

While partitioning is a straightforward concept, determining the best partitioning column(s) requires a solid understanding of how data is modified by the application. It is not recommended to partition the primary key due to cardinality concerns. Large tables that require partitioning are typically "transactional" in nature – e.g. sales data. Typically selecting a date column or adding a YEAR_MONTH column to the target data set within Replicate provides a good method for partitioning. Below is an example of partitioning's impact on cluster utilization and thus latency. In this example, a table of approximately 68 million source rows / 655 GB of data was processing a production CDC workload. Partitioning the delta table using a DATE column achieved a 73% reduction in latency and a large reduction in memory and CPU consumption on the cluster.

Cluster Utilization – Not Partitioned



Cluster Utilization – Partitioned



Appendix I - Creating Connections

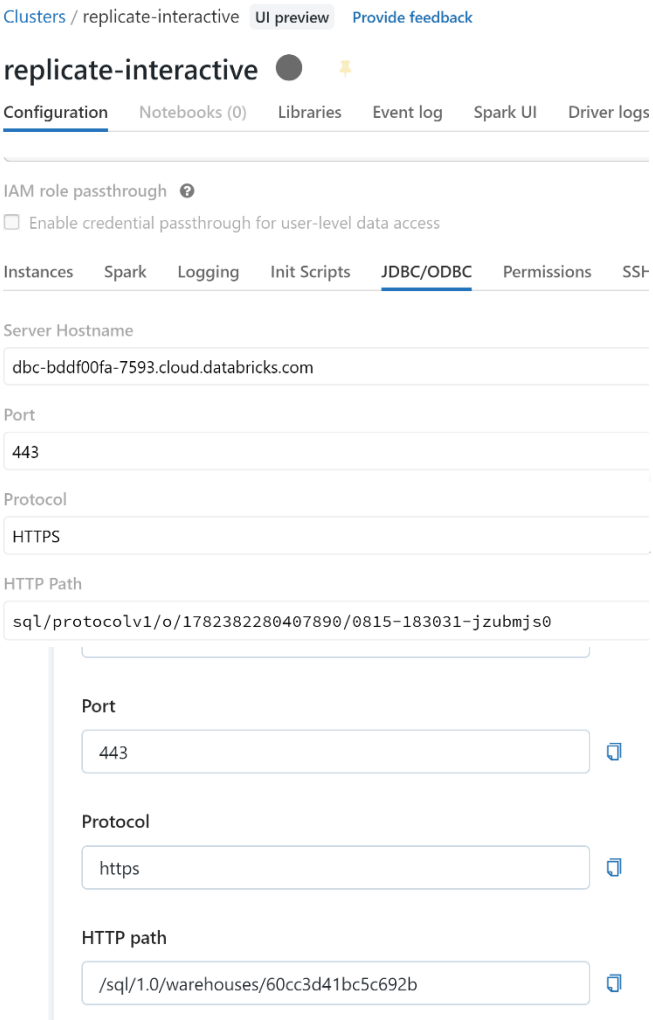
	It is highly recommended that you check the documentation available on help.qlik.com for the latest requirements and supported configurations for your Databricks environment

To create a data connection in Qlik Replicate to a Databricks instance you will need:

1. Server Hostname
2. Port
3. HTTP Path
4. Token

- Clusters

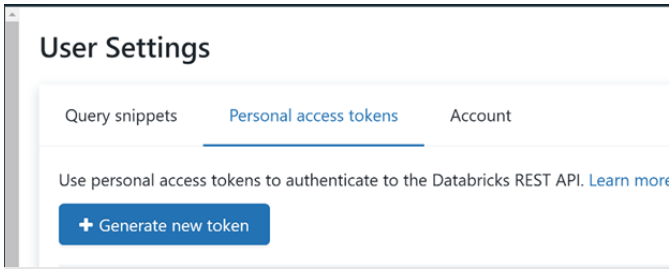
  Information 1, 2, and 3 can be extracted from the Databricks console by going to the cluster configuration or your SQL Endpoint configuration, and under the Advanced Options section you will find the JDBC/ODBC tab

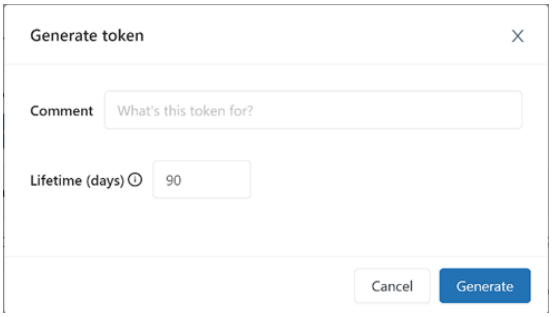


- Databricks SQL Warehouse

  Information 1, 2, and 3 can be extracted from the Databricks console by going to the SQL Warehouse section under the "Connection details" tab

To get an access token, you need to go to the User Settings section of your Databricks console and use the Generate new token button.





	It is important to store the generated token in a safe place because you cannot retrieve it again after you close this dialogue

Qlik transforms complex data landscapes into actionable insights, driving strategic business outcomes. Serving over 40,000 global customers, our portfolio leverages advanced, enterprise-grade AI/ML and pervasive data quality. We excel in data integration and governance, offering comprehensive solutions that work with diverse data sources. Intuitive and real-time analytics from Qlik uncover hidden patterns, empowering teams to address complex challenges and seize new opportunities. Our AI/ML tools, both practical and scalable, lead to better decisions, faster. As strategic partners, our platform-agnostic technology and expertise make our customers more competitive.

Qlik.com