ONEHOUSE

Onehouse Managed Lakehouse Table Optimizer Quick Start

May 14, 2024

Solution Overview

Onehouse provides integrated table services for optimizing read and write performance of Hudi tables. Compaction, clustering, and cleaning are supported as out-of-the-box features for Hudi tables managed within Onehouse as part of an ingestion stream capture. For more details, refer to the Onehouse Docs.

In addition, Apache Hudi users can now utilize Onehouse Lakehouse Table Optimizer to manage Hudi tables that are *not* created and operated by Onehouse. This quick start guide provides detailed instructions and prerequisites to setup and use Onehouse Lakehouse Table Optimizer.

Prerequisites

To begin, ensure you have a Onehouse account and have completed the onboarding process outlined in the Getting Started documentation.

As external writers/jobs interact with a Hudi table while Onehouse's managed table services optimize it, a shared lock provider is essential to coordinate access by multiple writers/jobs. The following shared lock providers are supported:

- Zookeeper: A Zookeeper instance (configured without authentication enabled) accessible from the Onehouse EKS cluster.

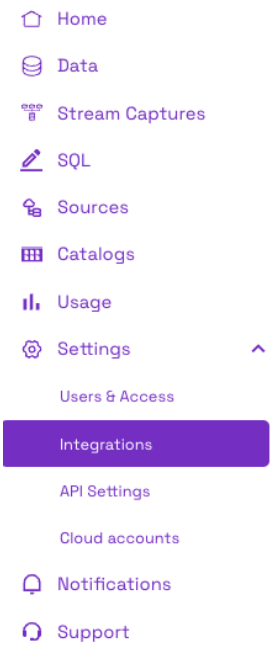
- DynamoDB: An Amazon DynamoDB table accessible from the Onehouse EKS cluster. This DynamoDB table will serve as the repository for shared locks.

Implementation Guide

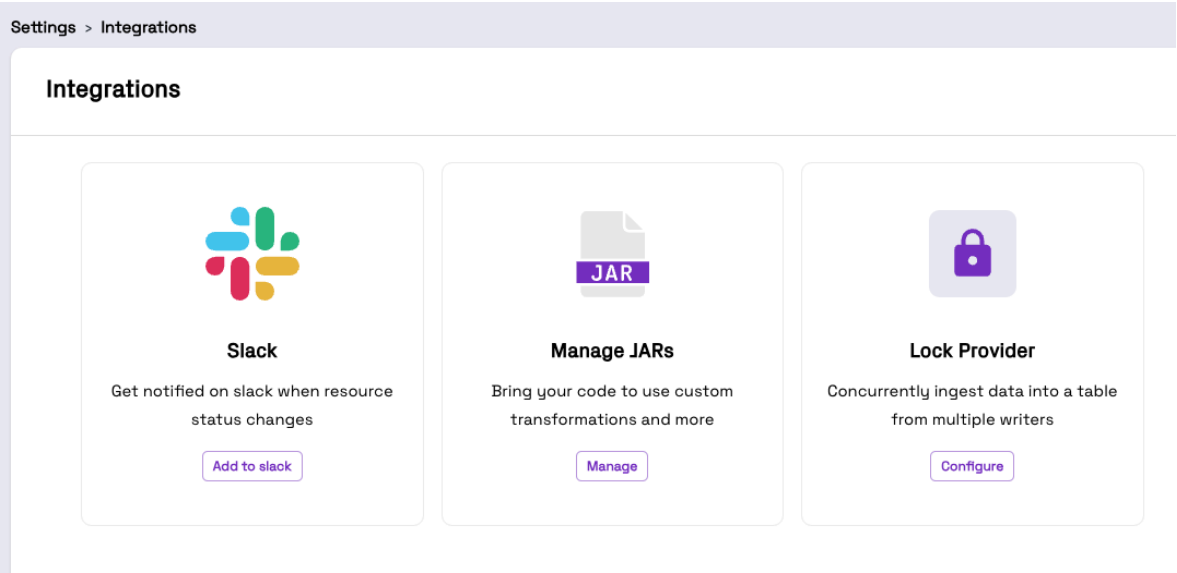
Log in to the Onehouse Console at cloud.onehouse.ai and follow:

Add a lock provider instance in Onehouse

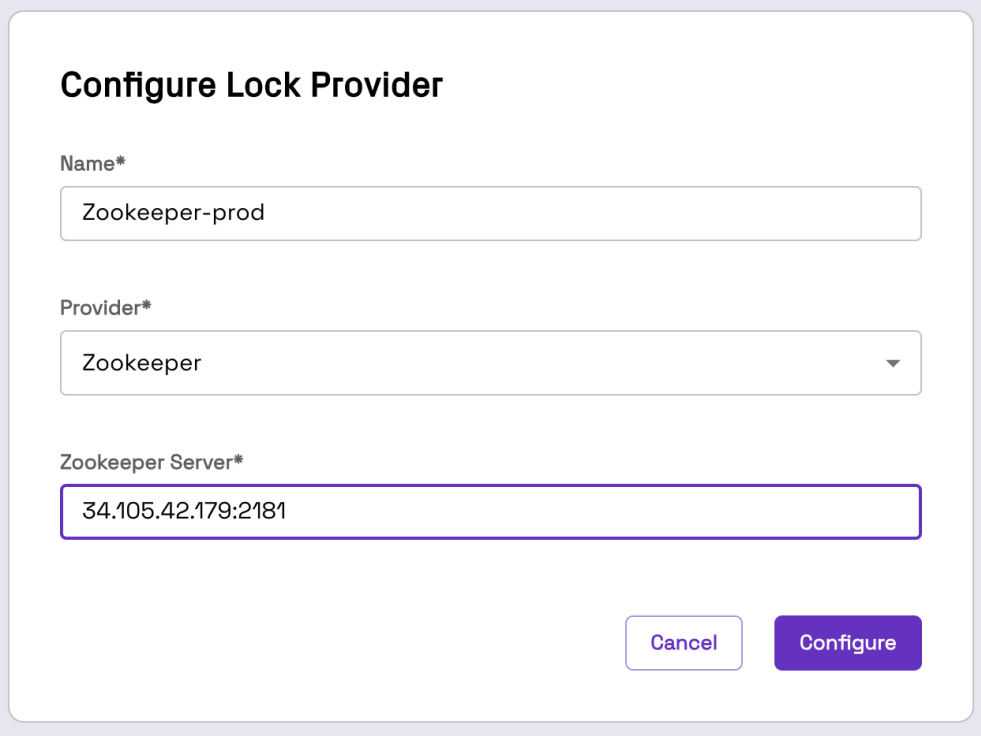
- Navigate to Settings -> Integrations.



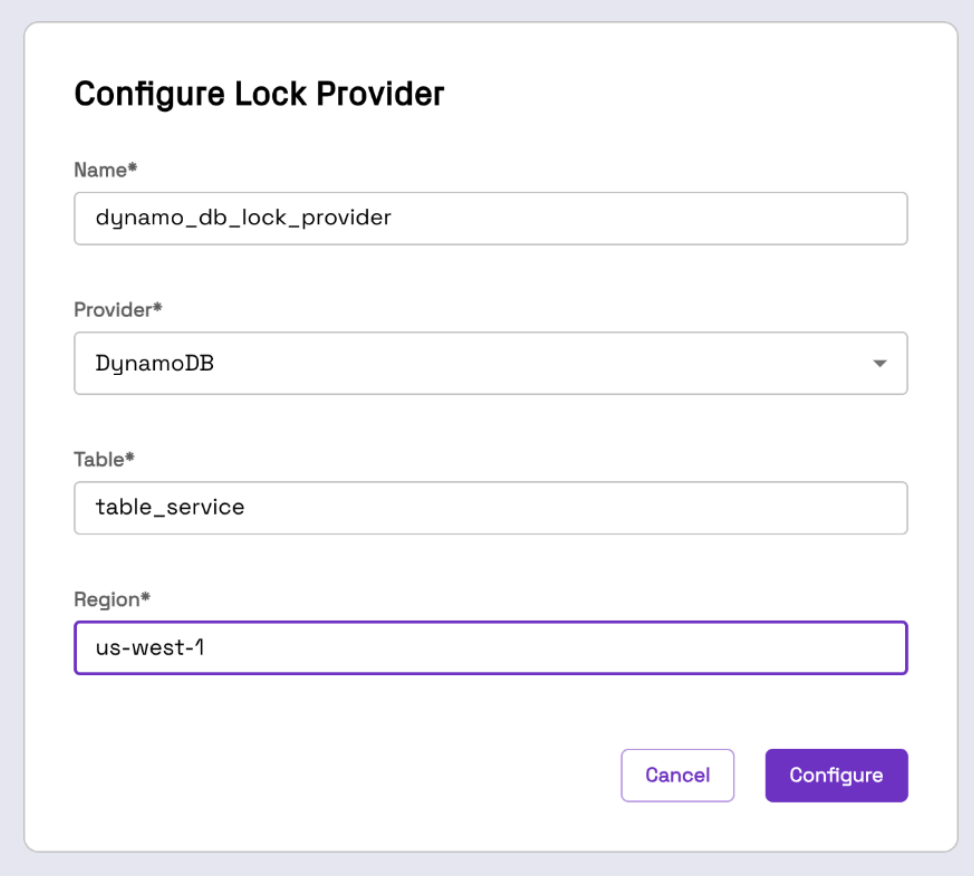
- If Concurrency Control is enabled for your Onehouse account, you will see a tile for 'Lock Provider' configuration. *Note: If this feature is not enabled in your Onehouse environment, please reach out to your Onehouse representative or email gtm@onehouse.ai.*
- Click 'Configure'.



- Provide a 'Name', select **Zookeeper/DynamoDB** for "Provider".
- For **Zookeeper**, provide a comma-separated list of host:port for 'Zookeeper Servers':



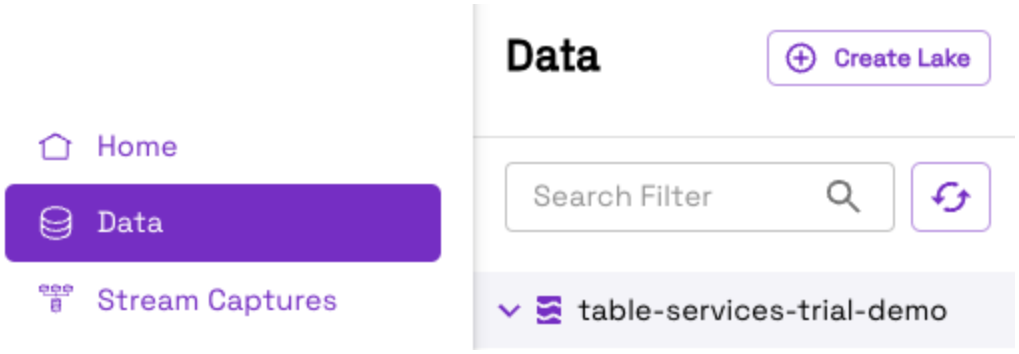
- For **DynamoDB**, enter a name for the lock provider and provide the DynamoDB table name and region.



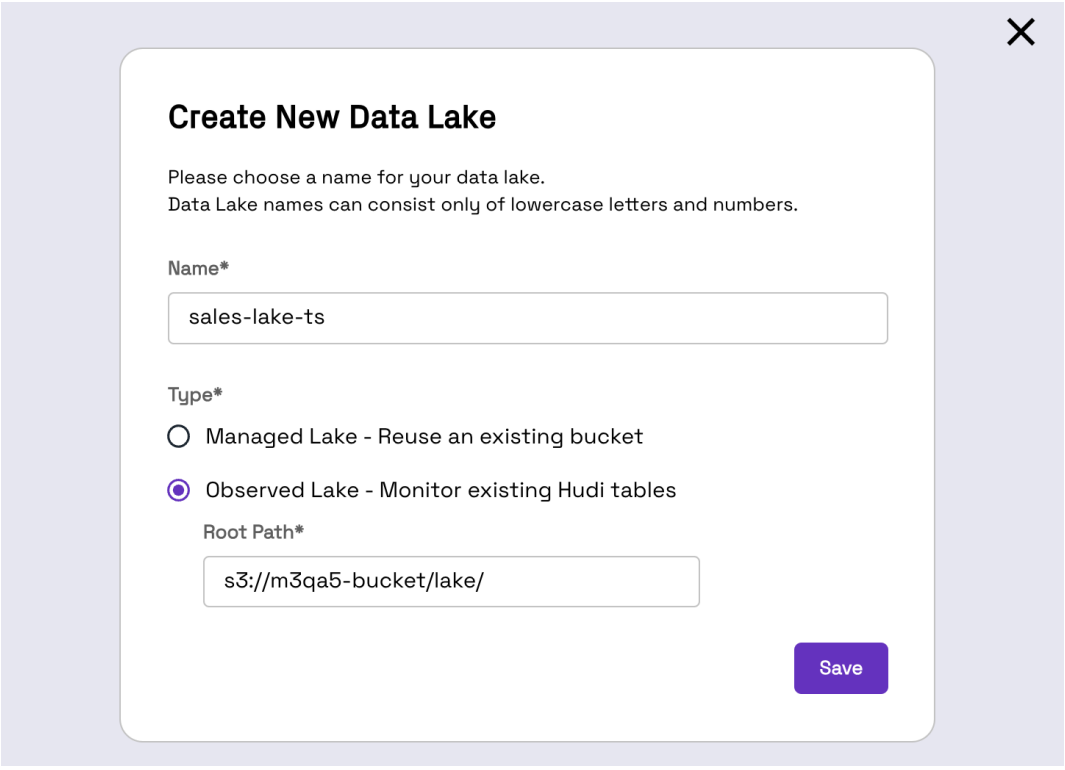
Note: Because Onehouse doesn't have the permissions required to create DynamoDB tables, you need to create one yourself. Make sure an attribute with the name "key" is present in the DynamoDB lock table. The key attribute should be the partition key and you don't have to specify the sort key. References: https://hudi.apache.org/docs/concurrency_control#amazon-dynamodb-based-lock-provider and https://hudi.apache.org/docs/configurations/#DynamoDB-based-Locks-Configurations.

Discover Hudi tables as Onehouse Observed Lake

- Navigate to 'Data' and click 'Create Lake'.



- Provide a 'Name' for the lake, select 'Observed Lake' for 'Type', and specify the S3 path prefix that contains the Hudi tables to be managed as the 'Root Path'. Click 'Save'.



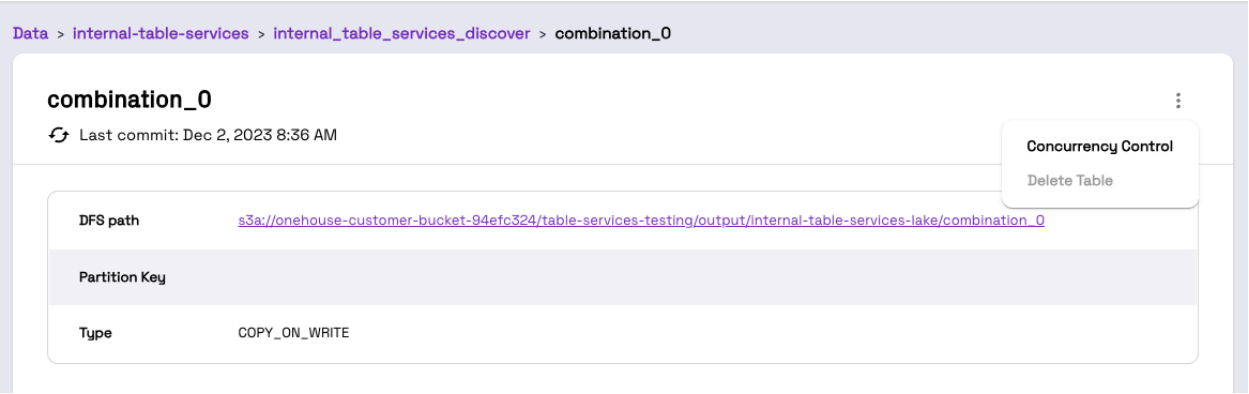
- All Hudi tables with the S3 path prefix provided in the last step will be discovered in Onehouse. Table discovery is expected to take a few minutes. After the initial

discovery, new tables are discovered by a sync operation which runs once every hour.

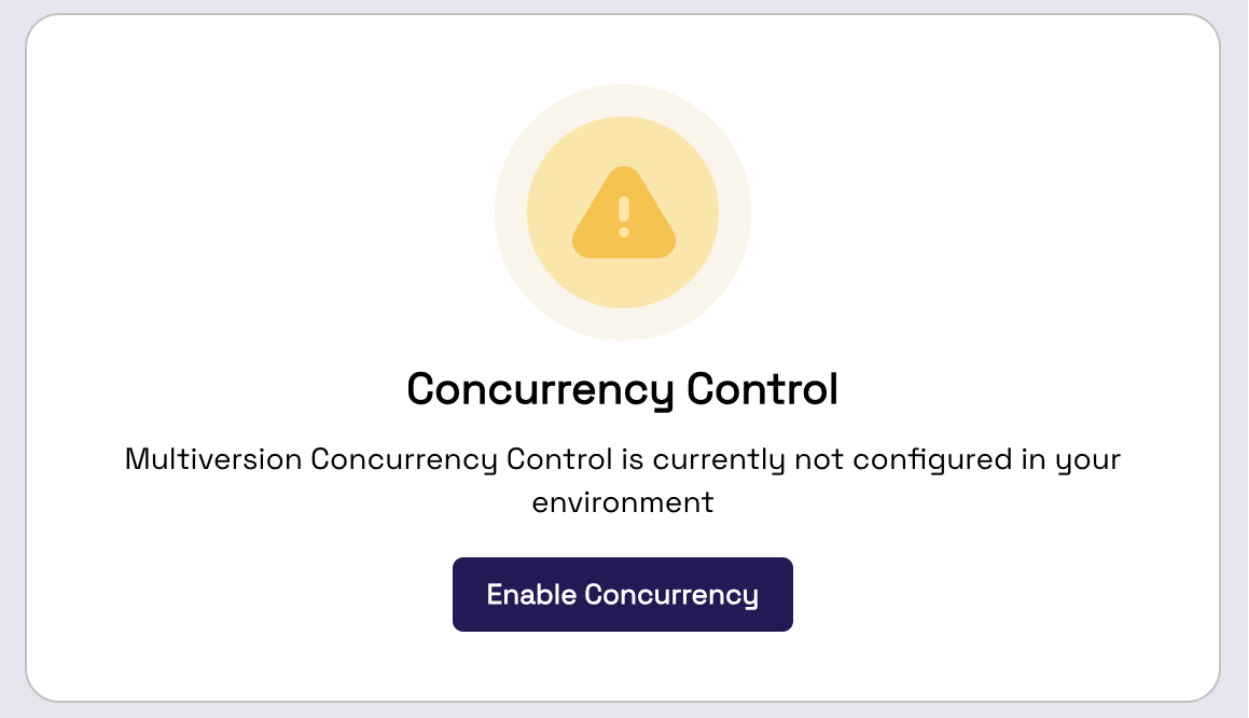
- Metadata for discovered tables is synced once every 5 minutes.
- ***Note: The S3 path prefix provided for Observed Lake creation must be unique across all lakes created in Onehouse.***

Enable Concurrency Control

- Navigate to one of the discovered Hudi tables and click 'Concurrency Control' from the table menu. This option is grayed if a lock provider is not configured.



- Click 'Enable Concurrency'.



Update External Writers and Jobs

- Use the Hudi properties shown below to update all external writers/jobs accessing this table outside Onehouse to enable concurrency control. This step must be completed before proceeding.

Concurrency Control

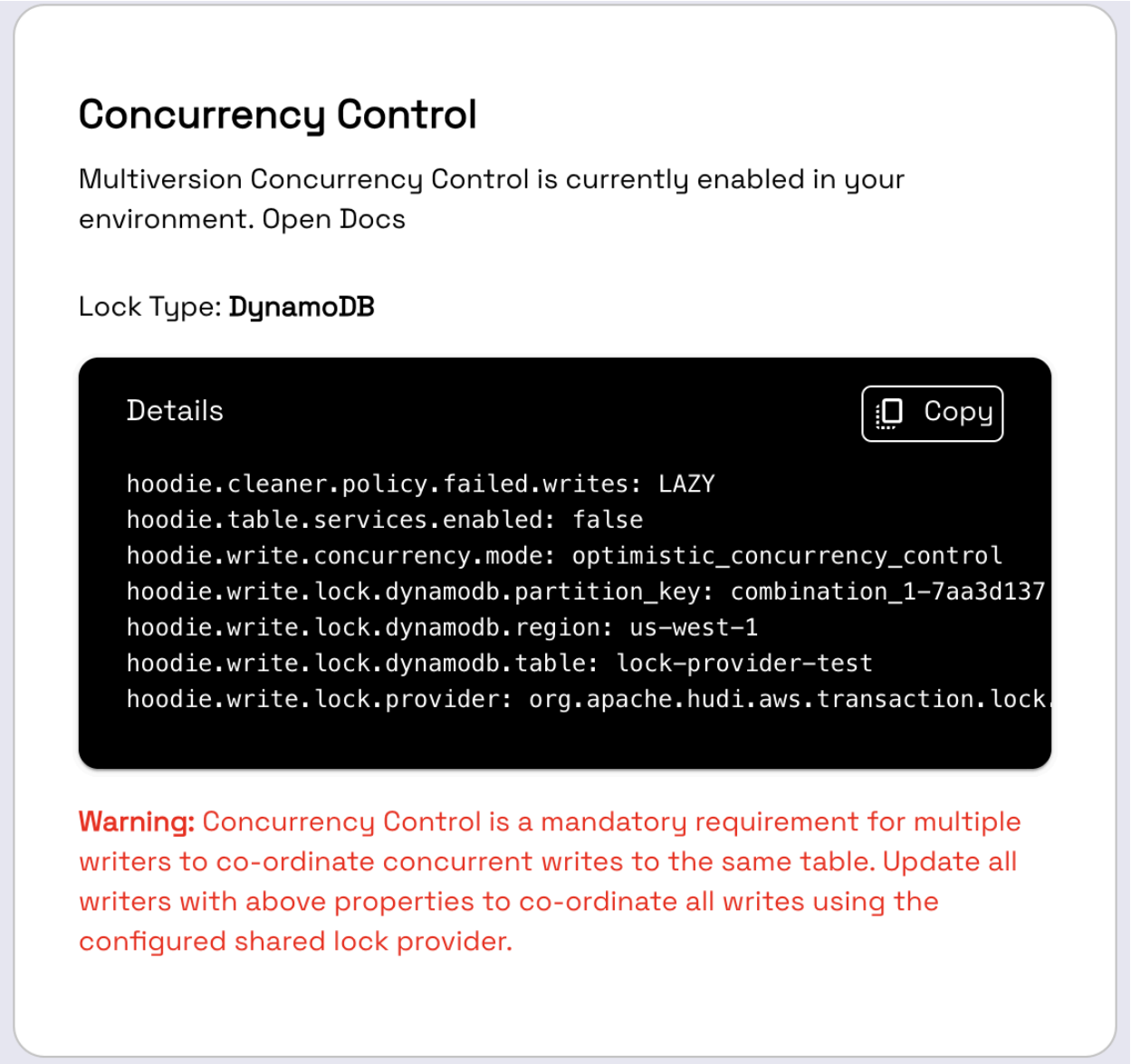
Multiversion Concurrency Control is currently enabled in your environment. Open Docs

Lock Type: **Zookeeper**

Details Copy

```
hoodie.cleaner.policy.failed.writes: LAZY
hoodie.table.services.enabled: false
hoodie.write.concurrency.mode: optimistic_concurrency_control
hoodie.write.lock.provider: org.apache.hudi.client.transaction.lock
hoodie.write.lock.zookeeper.base_path: /tmp/githubpullrequest_table_
hoodie.write.lock.zookeeper.lock_key: lock_key
hoodie.write.lock.zookeeper.port: 2181
hoodie.write.lock.zookeeper.url: 34.105.42.169
```

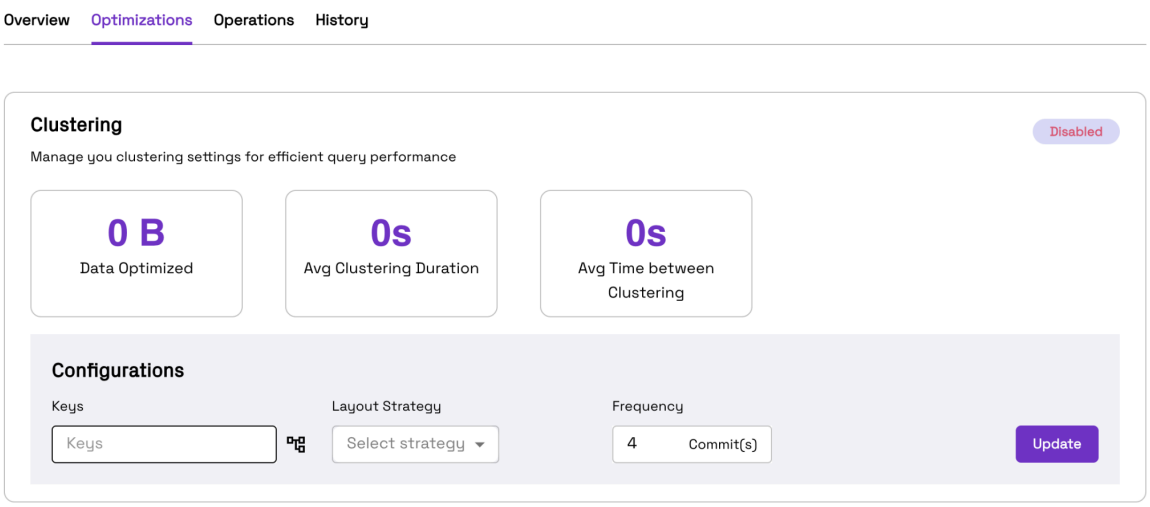
Warning: Concurrency Control is a mandatory requirement for multiple writers to co-ordinate concurrent writes to the same table. Update all writers with above properties to co-ordinate all writes using the configured shared lock provider.



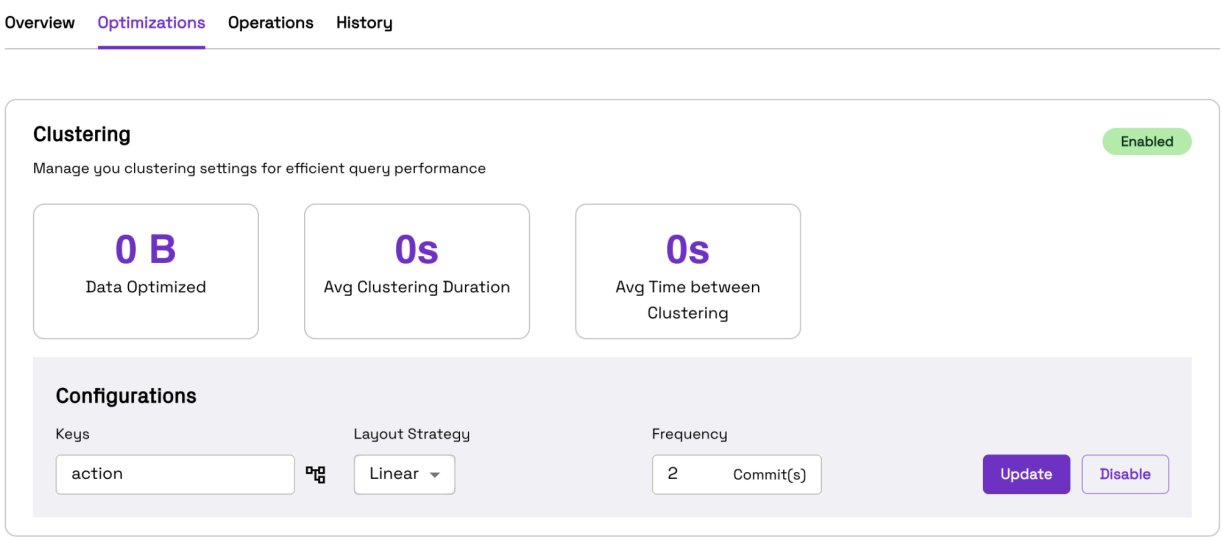
The partition_key is in the form of "<Hudi_table_name>-<a truncated UUID based on the s3 base path>". This is to ensure uniqueness per table and avoid name collisions.

Configure Table Services

- Once all external writers/jobs are updated to utilize concurrency, proceed to the 'Optimizations' tab for the Hudi table and configure the services according to the instructions provided in the Onehouse documentation.



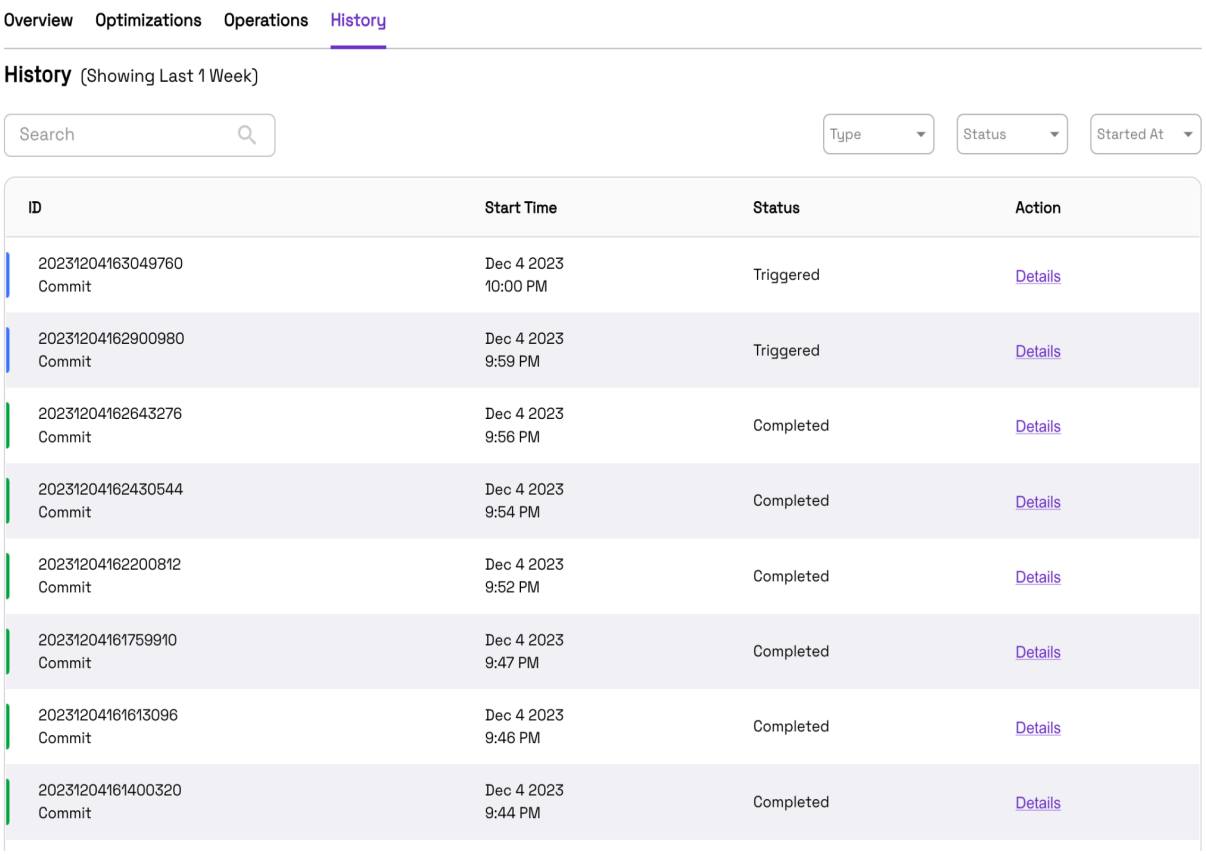
- Initially the clustering option is “Disabled”. Provide the keys or columns to sort the files (mandatory) and a table layout optimization strategy (default is Linear) and click “Update” to enable.



- Use the “Disable” button to stop the execution of a table service.

Monitoring of Table Service Jobs

Navigate to the ‘History’ tab to observe successful runs of table services jobs as well as commits made by external writers/jobs:



Conclusion

Onehouse Lakehouse Table Optimizer offers customers with self-managed Hudi pipelines a fully managed solution for running and monitoring table management services for their Hudi data lakehouses. This empowers customers to focus on crucial data ingestion and transformation tasks for analytics and ML, leading to improved performance. Additionally, it provides a user-friendly service for managing, maintaining, and optimizing Hudi tables and underlying Parquet files.

If you are ready to give Onehouse a try, or want to learn more, please visit the Onehouse listing on the AWS Marketplace or sign up for a Onehouse free trial.