

# Apache Iceberg Overview

Date published: 2022-07-24

Date modified: 2022-07-24

# CLOUDBERA

# Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

**Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.**

# Contents

**Iceberg overview..... 4**

## Iceberg overview

Apache Iceberg is a table format for huge analytics datasets that defines how metadata is stored and data files are organized. Iceberg is also a library that compute engines can use to read/write a table.

You can efficiently query large Iceberg tables on object stores. Iceberg supports concurrent reads and writes on all storage media. You can use Iceberg when a single table contains tens of petabytes of data, and you can read these tables without compromising performance. By default, Hive and Impala use the Iceberg HiveCatalog for Metastore management of Iceberg tables. The catalog stores Iceberg metadata, including the location of the table. Iceberg catalog use of Hive metastore (HMS) for catalog operations is lightweight. Iceberg relieves HMS pressure by storing partition information in metadata files on the file system/object store instead of within the HMS. This architecture supports rapid scaling without performance hits.

In Cloudera Public Cloud Data Hub, you can deploy Iceberg based applications across multiple clouds including AWS, Azure and Google Cloud.

### Key features overview

From Hive and Impala, you can read and write V1 or V2 Iceberg tables. The following features are included:

- Reads of Iceberg V2 tables that have had row-level deletes or updates makes Apache Iceberg ACID-compliant with serializable isolation and an optimistic concurrency model
- Materialized views of Iceberg tables
- Enhanced maintenance features, such as expiring and removing old snapshots and compaction of small files
- Performance and scalability enhancements

### Iceberg table security and visualization

Apache Iceberg integrates Apache Ranger for security. You can use Ranger integration with Hive and Impala to apply fine-grained access control to sensitive data in Iceberg tables. Iceberg is also integrated with Data Visualization for creating dashboards and other graphics of your Iceberg data.

### Supported ACID transaction properties

Iceberg supports atomic and isolated database transaction properties. Writers work in isolation, not affecting the live table, and perform an atomic swap for metadata pointers only when the transactions are committed, making the changes in one atomic commit.

Iceberg uses snapshots to guarantee isolated reads and writes. You see a consistent version of table data without locking the table. Readers always see a consistent version of the data without the need to lock the table. Writers work in isolation, not affecting the live table.

### Iceberg partitioning

The Iceberg partitioning technique has performance advantages over conventional partitioning, such as Apache Hive partitioning. Iceberg hidden partitioning is easier to use. Iceberg supports in-place partition evolution; to change a partition, you do not rewrite the entire table to add a new partition column, and queries do not need to be rewritten for the updated table. Iceberg continuously gathers data statistics, which supports additional optimizations, such as partition pruning.

Iceberg uses multiple layers of metadata files to find and prune data. Impala, Spark, and Flink keep track of data at the folder level and not at the file level, performing file list operations when working with data in a table. Performance problems occur during the execution of multiple directory listings. Iceberg keeps track of a complete list of files within a table using a persistent tree structure. Changes to an Iceberg table use an atomic object/file level commit to update the path to a new snapshot file. The snapshot points to the individual data files through manifest files.

The manifest files track data files across partitions, storing partition information and column metrics for each data file. A manifest list is an additional index for pruning entire manifests. File pruning increases efficiency.

Iceberg relieves Hive metastore (HMS) pressure by storing partition information in metadata files on the file system/object store instead of within the HMS. This architecture supports rapid scaling without performance hits.

### **Related Information**

[Iceberg support from Spark in Cloudera Data Engineering](#)

[Apache Software Foundation Iceberg Docs](#)

[Using Apache Iceberg](#)