# Migrating Hive and Impala Workloads to CDP

**Date published: 2023-08-14**
**Date modified: 2023-08-18**

## CLOUDERA

# Legal Notice

# Contents

# Migrating Hive and Impala workloads to CDP

You learn how to accelerate the migration process, to refactor Hive applications, and to handle semantic changes from Hive 1/2 to Hive 3. You get pointers to Impala documentation about workload migration and application refactoring.

## Handling prerequisites

You must perform a number of tasks before refactoring workloads.

You need to handle the following prerequisites:

- Identify all the Hive 1/2 workloads in the cluster (CDH 5.x, CDH 6.x, HDP 2.x, HDP 3.x). (This document assumes all the identified workloads are in working condition).
- Prepare tables for migration.

  To prepare the tables for migration, use the Hive SRE Tool which is a Cloudera Lab tool that scans your Hive Metastore and HDFS to identify common upgrade problems that can cause the CDP upgrade to fail. The tool provides guidance for fixing those problems before migrating the tables to CDP. This guidance is provided through reports that the administrator must take action on. The tool does not take corrective action itself.

  Cloudera strongly recommends running the Hive SRE Tool to aid in this pre-upgrade HMS healthcheck. If you do not run it, you must manually investigate your HMS for the following types of problems.

  Refer to Hive SRE Tooling for tool setup, tool execution & interpretation of output using Hive SRE Tool on Hive Metadata.

  The general transition from Hive 1 and 2 to Hive 3 includes the following types of HMS operations. The Hive SRE Tool performs the equivalent types of checks in an automated fashion. Please review these from the Cloudera documentation site (CDH 5, CDH 6, HDP 2, HDP 3).

The following table lists Hive table conditions, the impact of the conditions, and distributions affected. In this table, XXX stands for some value/count. This value is the number of affected paths you get from the Hive-SRE tool output reports. For each type of condition listed in the table, there is some number of paths/tables affected.

| Condition | Impact | Distributions Affected |
|---|---|---|
| Bad Filenames | Tables that would be converted from a Managed Non-Acid table to an ACID transactional table require the files to match a certain pattern. This process will scan the potential directories of these tables for bad filename patterns. When located, it will indicate which tables/partitions have file naming conventions that would prevent a successful conversion to ACID.<br><br>The best and easiest way to correct these file names is to use HiveSQL to rewrite the contents of the table/partition with a simple 'INSERT OVERWRITE TABLE xxx SELECT * FROM xxx'.<br><br>This type of statement will replace the current bad filenames with valid file names by rewriting the contents in HiveSQL.<br><br>There are approximately XXX paths that may need remediation. The list of paths can be found in the output of the Hive assessment report in the u3/bad_filenames_for_orc_conversion.md. | HDP 2 |

| Condition | Impact | Distributions Affected |
|---|---|---|
| Missing Directories | Missing Directories cause the upgrade conversion process to fail. This inconsistency is an indicator that data was removed from the file system, but the Hive MetaStore was not updated to reflect that operation. An example of this is deleting a partition in HDFS, without dropping the partition in Hive. <br><br> There are XXX affected paths that need remediation. The list of directories can be found in the output of the Hive assessment report, in the file u3/loc_scan_missing_dirs.md. | CDH5, CDH6, HDP2, HDP3 |
| Managed Shadow Tables | In Hive 3, Managed tables are 'ACID' tables. Sharing a location between two 'ACID' tables will cause compaction issues and data issues. These need to be resolved before the upgrade. <br><br> There are XXX affected tables that need remediation. The list of paths can be found in the output of the Hive assessment report, in the file u3/hms_checks.md | CDH5, CDH6, HDP2, HDP3 |
| Managed Table Migrations | This process will list tables that will and 'could' be migrated to "Managed ACID" tables during the upgrade process. <br><br> Tables used directly by Spark or if data is managed by a separate process that interacts with the FileSystem, you may experience issues post-upgrade. <br><br> Recommended: Consider converting these tables to external tables. <br><br> There are XXX affected tables that may need remediation. The list of tables can be found in the output of the Hive assessment report in the file managed_upgrade_2_acid.sql. | CDH5, CDH6, HDP2 |
| Compaction Check | In the upgrade to Hive 3, ACID tables must be compacted prior to initiating the upgrade. <br><br> XXX tables were noted as requiring compaction. Because CDH does not support Hive ACID tables, this may be a leftover from the previous HDP to CDH migration that the customer implemented. <br><br> This should be investigated further. The affected table is ers_stage_tls.test_delete | CDH5, CDH6, HDP2, HDP3 |
| Unknown SerDe Jars | A list tables using SerDe's that are not standard to the platform appears. Review the list of SerDes and verify they are still necessary and available for CDP. <br><br> There are approximately XXX tables configured with 3rd party SerDes. | CDH5, CDH6, HDP2 |

| Condition | Impact | Distributions Affected |
|---|---|---|
| Remove transactional=false from Table Properties | In CDH 5.x it is possible to create tables with the property transactional=false set. While this is a no-op setting, if any of your Hive tables explicitly set this, the upgrade process fails.<br><br>You must remove 'transactional'='false' from any tables you want to upgrade from CDH 5.x to CDP.<br><br>Alter the table as follows:<br><br>ALTER TABLE my_table UNSET TBLPROPERTIES ('transactional'); | CDH5, CDH6 |
| Make Tables SparkSQL Compatible | Non-Acid, managed tables in ORC or in a Hive Native (but non-ORC) format that are owned by the POSIX user hive will not be SparkSQL compatible after the upgrade unless you perform manual conversions.<br><br>If your table is a managed, non-ACID table, you can convert it to an external table using this procedure (recommended). After the upgrade, you can easily convert the external table to an ACID table, and then use the Hive Warehouse Connector to access the ACID table from Spark.<br><br>Take one of the following actions.<br><br>• Convert the tables to external Hive tables before the upgrade.<br><br>  ALTER TABLE ... SET TBLPROPERTIES('EXTERNAL'='TRUE', 'external.table.purge'='true')<br>• Change the POSIX ownership to an owner other than hive.<br><br>  You will need to convert managed, ACID v1 tables to external tables after the upgrade. | CDH5, CDH6, HDP2 |
| Legacy Kudu Serde Report | Early versions of Hive/Impala tables using Kudu were built before Kudu became an Apache Project. After Kudu became an Apache Project, the base Kudu Storage Handler classname changed. This report locates and reports on tables using the legacy storage handler class. | CDH5, CDH6, HDP2, HDP3 |
| Legacy Decimal Scale and Precision Check | When the DECIMAL data type was first introduced in Hive 1, it did NOT include a Scale or Precision element. This causes issues in later integration with Hive and Spark. You need to identify and suggest corrective action for tables where this condition exists. | CDH5, CDH6, HDP2, HDP3 |
| Database / Table and Partition Counts | Use this to understand the scope of what is in the metastore. | CDH5, CDH6, HDP2, HDP3 |
| Small Files, Table Volumes, Empty Datasets | Identify and fix these details to clean up unwanted datasets in the cluster which would speed up the Hive upgrade process. | CDH5, CDH6, HDP2, HDP3 |

| Condition | Impact | Distributions Affected |
|---|---|---|
| Merge Independent Hive and Spark Catalogs | In HDP 3.0 - 3.1.4, Spark and Hive use independent catalogs for accessing tables created using SparkSQL or Hive tables. A table created from Spark resides in the Spark catalog. A table created from Hive resides in the Hive catalog. Databases fall under the catalog namespace, similar to how tables belong to a database namespace. In HPD 3.1.5, Spark and Hive share a catalog in Hive metastore (HMS) instead of using separate catalogs.<br><br>The Apache Hive schematool in HDP 3.1.5 and CDP releases supports the mergeCatalog task. | HDP3 |

# Hive 1 and 2 to Hive 3 changes

A description of the change, the type of change, and the required refactoring provide the information you need for migrating from Hive 1 or 2 to Hive 3.

In addition to the topics in this section that describe Hive changes, see the following documentation about changes applicable to CDH and HDP to prepare the workloads.

- Hive Configuration Changes Requiring Consent
- Unsupported Interfaces and Features

## Reserved keywords

Reserved words (also called keywords) have a predefined meaning and syntax in Hive. These keywords have to be used to develop programming instructions. Reserved words cannot be used as identifiers for other programming elements, such as the name of variable or function.

Hive 1 and 2

TIME, NUMERIC, SYNC are not reserved keywords.

Hive 3

TIME, NUMERIC, SYNC are reserved keywords.

Action Required

Reserved keywords are permitted as identifiers if you quote them. As an example, if in scripts there are identifiers like TIME, NUMERIC, SYNC use backtick `` to quote it like, `TIME`, `NUMERIC`, `SYNC`.

Distribution Affected

CDH5, CDH6, HDP2

## Spark-client JAR requires prefix

The Spark-client JAR should be prefixed with hive- to make sure the jar name is consistent across all Hive jars.

Hive 1 and 2

You reference spark-client jar as spark-client.

Hive 3

You reference the spark-client jar as hive-spark-client.

Action Required

Change references as described above.

## Hive warehouse directory

Hive stores table files by default in the Hive Warehouse directory.

Hive 1 and 2

The warehouse directory path is /user/hive/warehouse (CDH) or /apps/hive/warehouse (HDP)

Hive 3

hte warehouse directory path for a managed table is /warehouse/tablespace/managed/hive and the default location for external table is /warehouse/tablespace/external/hive.

Action Required

As ACID tables reside by default in /warehouse/tablespace/managed/hive and only the Hive service can own and interact with files in this directory, applications accessing the managed tables directly need to change their behavior in Hive3.

As an example, consider the changes required for Spark application. As shown in the table below, if during upgrade Hive1/2 managed tables gets converted to an external table in Hive3, no refactoring is required for Spark application. If the Hive1/2 managed table gets converted to the managed table in Hive3, Spark application needs to refactor to use Hive Warehouse Connector (HWC) or the HWC Spark Direct Reader.

| Hive Table before upgrade in Hive1/2 | Hive Table after upgrade in Hive3 | Spark Refactoring |
|---|---|---|
| Managed | External | None |
| Managed | Managed | Use HWC or HWC Spark Direct Reader |
| External | External | None |
| | | |
| | | |
| | | |
| | | |

Distribution Affected

CDH5, CDH6, HDP2

## Replace Hive CLI with Beeline

Beeline is the new default command line SQL interface for Hive3.

Hive 1 and 2

Hive CLI is deprecated.

Hive 3

The original hive CLI is removed and becomes a wrapper around beeline.

Action Required

Remove, replace and test all hive CLI calls with the beeline command line statements.

For more information, see Converting Hive CLI scripts to Beeline.

Distribution Affected

CDH5, CDH6, HDP2, HDP3

## PARTIALSCAN

You need to identify and apply configuration-level changes, including removing the PARTIALSCAN option

Hive 1 and 2

ANALYZE TABLE ... COMPUTE STATISTICS supports the PARTIALSCAN option.

Hive 3

ANALYZE TABLE ... COMPUTE STATISTICS does not support PARTIALSCAN, which is retired and throws error.

For example:

```
ANALYZE TABLE test_groupby COMPUTE STATISTICS PARTIALSCAN;
Error: Error while compiling statement: FAILED: ParseException line 1:46
extraneous input 'PARTIALSCAN' expecting EOF near '<EOF>' (state=42000,code=
40000)
```

Action Required

Remove statements containing ALTER TABLE ... COMPUTE STATISTICS PARTIALSCAN. For example:

```
ANALYZE TABLE test_groupby COMPUTE STATISTICS PARTIALSCAN;
Error: Error while compiling statement: FAILED: ParseException line 1:46
extraneous input 'PARTIALSCAN' expecting EOF near '<EOF>' (state=42000,code=
40000)
```

Distribution Affected

CDH5, CDH6, HDP2.

## Concatenation of an external table

If the table or partition contains many small RCFiles or ORC files, then ALTER TABLE table_name [PARTITION (partition_key = 'partition_value' [, ...])] CONCATENATE will merge them into larger files. In the case of RCFile the merge happens at block level whereas for ORC files the merge happens at stripe level thereby avoiding the overhead of decompressing and decoding the data

Hive 1 and 2

You can concatenate an external table. For example:

```
ALTER TABLE table_name [PARTITION (partition_key = 'partition_value' [, ...]
)] CONCATENATE
```

Hive 3

Concatenation of an external table using CONCATENATE is not supported. For example, you get the following error:

```
alter table t6 concatenate;
Error: Error while compiling statement: FAILED: SemanticException [Error 300
34]: Concatenate/Merge can only be performed on managed tables (state=42000,
code=30034)
```

Action Required

Remove the CONCATENATE operation on external tables.

Distribution Affected

CDH5, CDH6

## INSERT OVERWRITE

INSERT OVERWRITE from a source with UNION ALL is not all for some tables.

Hive 1 and 2

For managed tables, INSERT OVERWRITE from a source with UNION ALL is allowed.

Hive 3

INSERT OVERWRITE from a source with UNION ALL on full CRUD ACID tables is not allowed.

For example, test_groupby_3 is a managed table and insert overwrite with union all will throw an error.

```
insert overwrite table test_groupby_3
. . . . . . . . . . . . . . . . . . . . . . . .> select col1,count from test
_groupby
. . . . . . . . . . . . . . . . . . . . . . .> union all
. . . . . . . . . . . . . . . . . . . . . . .> select col1,count from test
_groupby_1;

ERROR : FAILED: Error in acquiring locks: QueryId=hive_20220321114459_851b
1546-ca9f-4643-b888-7df23adaec66 is not supported due to OVERWRITE and UNION
 ALL.  Please use truncate + insert
```

Action Required

Instead of overwrite, use truncate and insert.

Distribution Affected

CDH5, CDH6, HDP2

## Managed to external table

In Hive1/2, a managed table can be converted to an external table by setting the external property of the table as true. In Hive3 this is restricted.

Hive 1 and 2

ALTER TABLE table_name SET TBLPROPERTIES('EXTERNAL'='true') can be performed on managed tables to convert external tables.

For example:

```
ALTER TABLE test_groupby_3 SET TBLPROPERTIES('EXTERNAL'='true')
ERROR : Failed
org.apache.hadoop.hive.ql.metadata.HiveException: Unable to alter table.
default.test_groupby_3 cannot be converted to external table as it is transa
ctional table.
```

Hive 3

ALTER TABLE table_name SET TBLPROPERTIES('EXTERNAL'='true') cannot be performed on managed tables to convert external tables.

Action Required

Remove a table property that sets EXTERNAL=true for a managed table.

Distribution Affected

CDH5, CDH6, HDP2

# Property changes affecting ordered or sorted subqueries and views

Order by/sort by without limit in subqueries is not supported in Hive 3.

Hive 1 and 2

You can use order by/sort by functionality in subqueries.

For example, the optimizer orders a subquery of the following table:

```
+--------------------+--------------------+
```

```
| test_groupby.col1  | test_groupby.col2   |
+--------------------+---------------------+
| test               | 2                   |
| test               | 3                   |
| test1              | 5                   |
+--------------------+---------------------+

select * from (select * from test_groupby order by col2 desc) t2;

+----------+-----------+
| t2.col1  | t2.col1   |
+----------+-----------+
| test1    | 5         |
| test     | 3         |
| test     | 2         |
+----------+-----------+
```

Hive 3

The optimizer removes order by/sort by without limit in subqueries and views. For example:

```
select * from (select * from test_groupby order by col2 desc) t2;

+----------+-----------+
| t2.col1  | t2.col1   |
+----------+-----------+
| test     | 2         |
| test     | 3         |
| test1    | 5         |
+----------+-----------+
```

Action Required

If the outer query has to perform some functional logic based on order of a subquery, the following query in Hive 2 returns a different result from Hive 3:

```
select col1 from (select * from test_groupby order by col2t desc ) t2 limit
1;
```

You must rewrite the query or set the following property to restore previous behavior:

```
set hive.remove.orderby.in.subquery=False
```

Distribution Affected

CDH5, CDH6

## Runtime configuration changes

There are a number of runtime configurations that Hive 3 does not support.

Remove unsupported configurations if explicitly set in scripts.

The following configurations supported in Hive 1 or 2 have been removed from Hive 3 and are not supported:

- hive.limit.query.max.table.partition
- hive.warehouse.subdir.inherit.perms
- hive.stats.fetch.partition.stats

Development of an HBase metastore for Hive started in release 2.0.0, but the work has stopped and the code was removed from Hive 3.0.0. The following Hive Metastore HBase configurations have been removed and are not supported:

- hive.metastore.hbase.cache.ttl
- hive.metastore.hbase.catalog.cache.size
- hive.metastore.hbase.aggregate.stats.cache.size
- hive.metastore.hbase.aggregate.stats.max.partitions
- hive.metastore.hbase.aggregate.stats.false.positive.probability
- hive.metastore.hbase.aggregate.stats.max.variance
- hive.metastore.hbase.cache.ttl
- hive.metastore.hbase.cache.max.writer.wait
- hive.metastore.hbase.cache.max.reader.wait
- hive.metastore.hbase.cache.max.full
- hive.metastore.hbase.catalog.cache.size
- hive.metastore.hbase.aggregate.stats.cache.size
- hive.metastore.hbase.aggregate.stats.max.partitions
- hive.metastore.hbase.aggregate.stats.false.positive.probability
- hive.metastore.hbase.aggregate.stats.max.variance
- hive.metastore.hbase.cache.ttl
- hive.metastore.hbase.cache.max.full
- hive.metastore.hbase.cache.clean.until
- hive.metastore.hbase.connection.class
- hive.metastore.hbase.aggr.stats.cache.entries
- hive.metastore.hbase.aggr.stats.memory.ttl
- hive.metastore.hbase.aggr.stats.invalidator.frequency
- hive.metastore.hbase.aggr.stats.hbase.ttl

# Prepare Hive tables for migration

To prepare the tables for migration, use Hive SRE Tool which is a Cloudera Lab tool that scans your Hive Metastore and HDFS to identify common upgrade problems that can cause the CDP upgrade to fail. The tool provides guidance for fixing those problems before migrating the tables to CDP. This guidance is provided through reports that the administrator must take action on. The tool does not take corrective action itself.

Cloudera strongly recommends running the Hive SRE Tool to aid in this pre-upgrade HMS healthcheck. If you do not run it, you must manually investigate your HMS for the following types of problems.

Please refer to Hive SRE Tooling for tool setup, tool execution & interpretation of output using Hive SRE Tool on Hive Metadata.

The general transition from Hive 1 and 2 to Hive 3 includes the following types of HMS operations. The Hive SRE Tool performs the equivalent types of checks in an automated fashion. Please review these from the Cloudera documentation site (CDH 5, CDH 6, HDP 2, HDP 3).

Note for section below:XXX stands for some value/count, this is a value of the number of affected paths we will get from the Hive-SRE tool output reports. For each type of condition listed in the table , there will be some number of paths/tables affected.

| Condition | Impact | Distribution Affected |
|---|---|---|
| Bad Filenames | Tables that would be converted from a Managed Non-Acid table to an ACID transactional table require the files to match a certain pattern. This process will scan the potential directories of these tables for bad filename patterns. When located, it will indicate which tables/partitions have file naming conventions that would prevent a successful conversion to ACID.<br><br>The best and easiest way to correct these file names is to use HiveSQL to rewrite the contents of the table/partition with a simple 'INSERT OVERWRITE TABLE xxx SELECT * FROM xxx'.<br><br>This type of statement will replace the current bad filenames with valid file names by rewriting the contents in HiveSQL.<br><br>There are approximately XXX paths that may need remediation. The list of paths can be found in the output of the Hive assessment report in the u3/bad_filenames_for_orc_conversion.md. | |
| Missing Directories | Missing Directories cause the upgrade conversion process to fail. This inconsistency is an indicator that data was removed from the file system, but the Hive MetaStore was not updated to reflect that operation. An example of this is deleting a partition in HDFS, without dropping the partition in Hive.<br><br>There are XXX affected paths that need remediation. The list of directories can be found in the output of the Hive assessment report, in the file u3/loc_scan_missing_dirs.md. | |
| Managed Shadow Tables | In Hive 3, Managed tables are 'ACID' tables. Sharing a location between two 'ACID' tables will cause compaction issues and data issues. These need to be resolved before the upgrade.<br><br>There are XXX affected tables that need remediation. The list of paths can be found in the output of the Hive assessment report, in the file u3/hms_checks.md | |
| Managed Table Migrations | This process will list tables that will and 'could' be migrated to "Managed ACID" tables during the upgrade process.<br><br>Tables used directly by Spark or if data is managed by a separate process that interacts with the FileSystem, you may experience issues post-upgrade.<br><br>Recommended: Consider converting these tables to external tables.<br><br>There are XXX affected tables that may need remediation. The list of tables can be found in the output of the Hive assessment report in the file managed_upgrade_2_acid.sql. | |

| Condition | Impact | Distribution Affected |
|---|---|---|
| Compaction Check | In the upgrade to Hive 3, ACID tables must be compacted prior to initiating the upgrade.<br><br>XXX tables were noted as requiring compaction. Because CDH does not support Hive ACID tables, this may be a leftover from the previous HDP to CDH migration that the customer implemented.<br><br>This should be investigated further. The affected table is ers_stage_tls.test_delete | |
| Unknown SerDe Jars | Will list tables using SerDe's that are not standard to the platform. Review list of SerDes and verify they are still necessary and available for CDP.<br><br>There are approximately XXX tables configured with 3rd party SerDes. | |
| Remove transactional=false from Table Properties | In CDH 5.x it is possible to create tables with the property   transactional=false set. While this is a no-op setting, if any of your Hive tables explicitly set this, the upgrade process fails.<br><br>You must remove 'transactional'='false' from any tables you want to upgrade from CDH 5.x to CDP.<br><br>Alter the table as follows:<br><br>ALTER TABLE my_table UNSET TBLPR OPERTIES   ('transactional'); | |
| Make Tables SparkSQL Compatible | Non-Acid, managed tables in ORC or in a Hive Native (but non-ORC) format that are owned by the POSIX user hive will not be SparkSQL compatible after the upgrade unless you perform manual conversions.<br><br>If your table is a managed, non-ACID table, you can convert it to an external table using this procedure (recommended). After the upgrade, you can easily convert the external table to an ACID table, and then use the Hive Warehouse Connector to access the ACID table from Spark.<br><br>Take one of the following actions.<br><br>• Convert the tables to external Hive tables before the upgrade.<br><br>`ALTER TABLE ... SET`<br>`    TBLPROPERTIES('EXTERNAL'='TRUE','external.table.purge'='true')`<br><br>• Change the POSIX ownership to an owner other than hive.<br><br>  You will need to convert managed, ACID v1 tables to external tables after the upgrade. | |

| Condition | Impact | Distribution Affected |
|---|---|---|
| Legacy Kudu Serde Report | Early versions of Hive/Impala tables using Kudu were built before Kudu became an Apache Project. Once it became an Apache Project, the base Kudu Storage Handler classname changed. This report locates and reports on tables using the legacy storage handler class. | |
| Legacy Decimal Scale and Precision Check | When the DECIMAL data type was first introduced in Hive 1, it did NOT include a Scale or Precision element. This causes issues in later integration with Hive and Spark. We'll identify and suggest corrective action for tables where this condition exists. | |
| Database / Table and Partition Counts | Use this to understand the scope of what is in the metastore. | |
| Small Files, Table Volumes, Empty Datasets | Identify and fix these details to clean up unwanted datasets in the cluster which would speed up the Hive upgrade process. | |
| Merge Independent Hive and Spark Catalogs | In HDP 3.0 - 3.1.4, Spark and Hive use independent catalogs for accessing tables created using SparkSQL or Hive tables. A table created from Spark resides in the Spark catalog. A table created from Hive resides in the Hive catalog. Databases fall under the catalog namespace, similar to how tables belong to a database namespace. In HPD 3.1.5, Spark and Hive share a catalog in Hive metastore (HMS) instead of using separate catalogs. The Apache Hive schematool in HDP 3.1.5 and CDP releases supports the mergeCatalog task. | |

## Impala changes from CDH to CDP

Cloudera documentation provides details about changes in Impala when migrating from CDH 5.13-5.16 or CDH 6.1 or later to CDP.

- Impala changes in CDP:This document lists down the syntax, service, property and configuration changes that affect Impala after upgrade from CDH 5.13-5.16 or CDH 6.1 or later to CDP.
- Change location of Datafiles: This document explains the impact on Hive warehouse directory location for Impala managed and external tables.
- Set Storage Engine ACLs : This document describes the steps to set ACLs for Impala to allow Impala to write to the Hive Warehouse Directory.
- Automatic Invalidation/Refresh of Metadata: In CDP, HMS and file system metadata is automatically refreshed. This document provides the details of change in behavior for invalidation and refresh of metadata between CDH and CDP.
- Metadata Improvements: In CDP, all catalog metadata improvements are enabled by default. This document lists down the metadata properties in CDP that can be customized for better performance and scalability.
- Default Managed Tables: In CDP, managed tables are transactional tables with the insert_only property by default. This document describes modifying file systems on a managed table in CDP and the methods to switch to the old CDH behavior.
- Automatic Refresh of Tables on Impala Clusters: In CDP tables or partitions are refreshed automatically on other impala clusters, this document describes the property that enables this behavior.
- Interoperability between Hive and Impala:This document describes the changes made in CDP for the optimal interoperability between Hive and Impala.

- ORC Support Disabled for Full-Transactional Tables: In CDP 7.1.0 and earlier versions, ORC table support is disabled for Impala queries. However, you have an option to switch to the CDH behavior by using the command line argument ENABLE_ORC_SCANNER.
- Metadata Improvements: In CDP, all catalog metadata improvements are enabled by default. This document lists down the metadata properties in CDP that can be customized for better performance and scalability.
- Default Managed Tables: In CDP, managed tables are transactional tables with the insert_only property by default. This document describes modifying file systems on a managed table in CDP and the methods to switch to the old CDH behavior.
- Automatic Refresh of Tables on Impala Clusters: In CDP tables or partitions are refreshed automatically on other impala clusters, this document describes the property that enables this behavior.
- Interoperability between Hive and Impala:This document describes the changes made in CDP for the optimal interoperability between Hive and Impala.
- ORC Support Disabled for Full-Transactional Tables: In CDP 7.1.0 and earlier versions, ORC table support is disabled for Impala queries. However, you have an option to switch to the CDH behavior by using the command line argument ENABLE_ORC_SCANNER.
- Authorization Provider for Impala:In CDP, Ranger is the authorization provider instead of Sentry. This document describes changes about Ranger enforcing a policy which may be different from using Sentry.
- Data Governance Support by Atlas:As part of upgrading a CDH cluster to CDP, Atlas replaces Cloudera Navigator Data Management for your cluster. This document describes the difference between two environments.

# Impala configuration differences in CDH and CDP

There are some configuration differences related to Impala in CDH and CDP. These differences are due to the changes made in CDP for the optimal interoperability between Hive and Impala and an improved user experience.

Review the changes before you migrate your Impala workload from CDH to CDP.

- Config changes:This document describes the default value changes and new configurations between CDH and CDP.
- Default File Formats: This document describes the change in default file format of the table in CDP and the steps to switch back to CDH behavior.
- Reconnect to HS2 Session: This document describes the behavior change in CDP for client connection to Impala and HS2.
- Automatic Row Count Estimation: This document describes the new default behavior in CDP for calculating statistics of the table and the property to switch back to CDH behavior.
- Using Reserved Words in SQL Queries:CDP follows ANSI SQL compliance, in which reserved words in SQL queries are rejected. This document provides details about reserved keywords and the method to use them.
- Other Miscellaneous Changes in Impala:This document describes other miscellaneous changes related to Impala Syntax and services that affect Impala during migration.

# Additional documentation

Cloudera documentation can help you migrate your Hive workloads.

For more information about migrating Hive workload, see the following documentation:

- Migrating tables to CDP
- Migrating Hive workloads from CDH
- Migrating Hive Workloads from HDP 2.6.5 after an in-place upgrade
- Replicating Hive data from HDP 3 to CDP
- Migrating Hive workloads to ACID

- Apache Hive Changes in CDP capture additional changes that need to be considered while upgrading from Hive 1/2 to Hive 3. The following documents summarize these changes:
  - Hive Configuration Property Changes
  - LOCATION and MANAGEDLOCATION clauses
  - Handling table reference syntax
  - Identifying semantic changes and workarounds
  - Unsupported Interfaces and Features
  - Changes to CDH Hive Tables
  - Changes to HDP Hive tables
- CDP Upgrade and Migrations Paths
- Migrating Hive 1-2 to Hive 3
- Migrating Hive Workloads from CDH
- SRE Tool
- Apache Hive Language Manual
- Release Notes