..

Migration paths from HDP 3 to CDP for LLAP users

Date published: 2022-02-04
Date modified:

Legal Notice

Contents

Migration paths from HDP 3 to CDP for LLAP users

If you are running your Hive HDP 3.x workloads using LLAP (low-latency analytical processing), you need to decide on the best migration path to CDP without compromising on the performance offered by LLAP. Migration recommendations depend on a number of factors, such as your workload type and whether you have Hive or Spark users.

Migration paths for Hive users

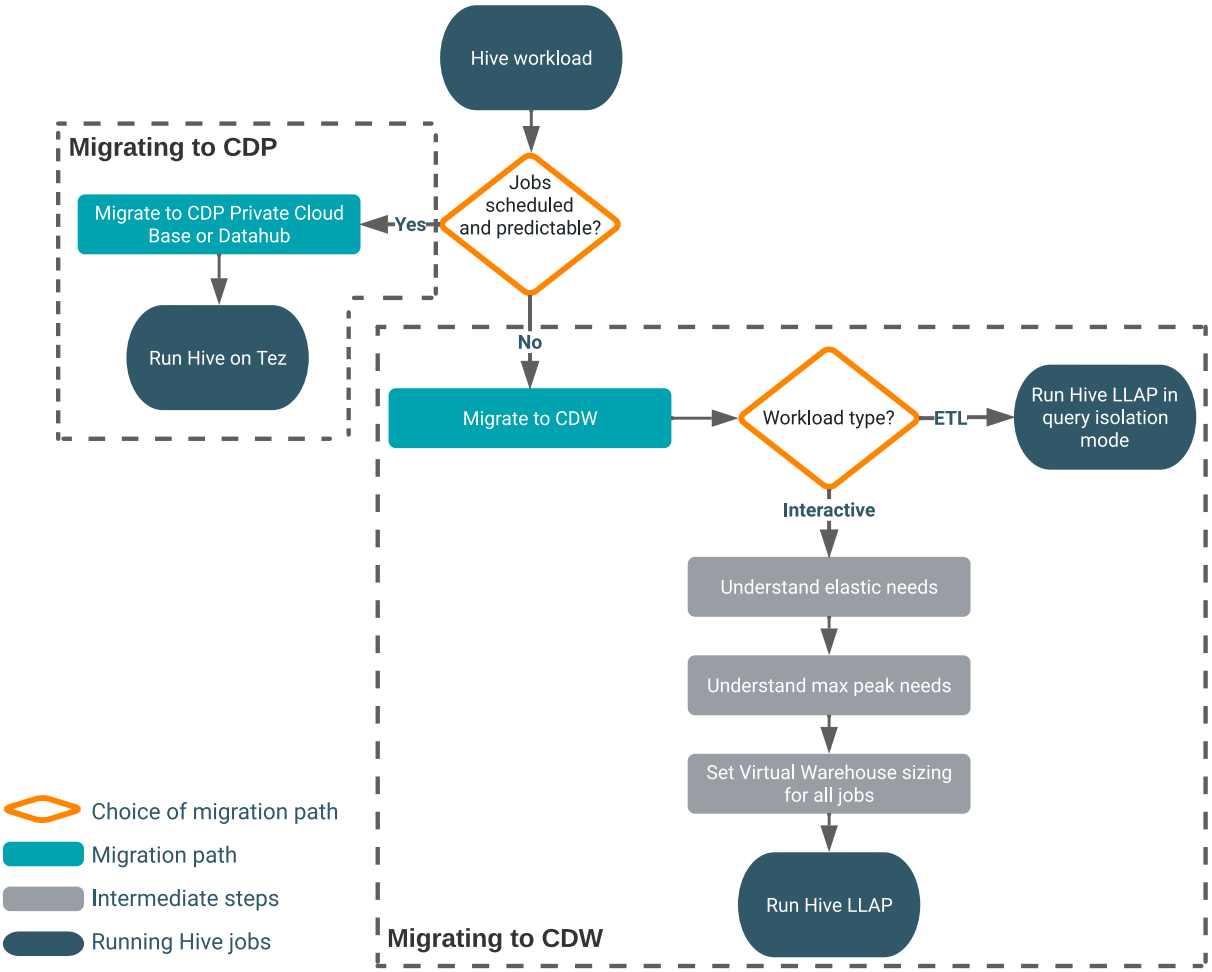
If you are on HDP and executing Hive workloads in Hive LLAP mode and want to upgrade to CDP, you can follow the migration path that matches your use case.

Using LLAP is recommended for running complex ETL jobs that are unpredictable. LLAP is offered as part of Cloudera Data Warehouse (CDW) and is not available in CDP Public Cloud (Data Hub) and CDP Private Cloud Base. Using Hive on Tez (no LLAP) is recommended for running jobs that are scheduled and predictable. Hive on Tez is supported on CDP Public Cloud and CDP Private Cloud Base.

The following migration paths are recommended based on how predictable your jobs are and your workload type:

- Migrate to CDP Public Cloud (Data Hub) or CDP Private Cloud Base — If your jobs are scheduled and predictable.
- Migrate to Cloudera Data Warehouse (CDW) — If your jobs are unpredictable and if they could increase demand on compute resources.

The following diagram shows these recommended paths:



Migration to CDP Private Cloud Base or CDP Public Cloud

If your Hive jobs are scheduled, or otherwise considered predictable, consider migrating to CDP Private Cloud Base or CDP Public Cloud and running your Hive workloads on Hive on Tez.

You learn why you should consider the predictability of your jobs when choosing a migration path from HDP 3 to CDP. If you were using LLAP on HDP, a path to CDP can offer equivalent performance and might make sense cost-wise.

LLAP is not supported in CDP Private Cloud Base and CDP Public Cloud, but Hive-on-Tez performs well for jobs that are not subjected to unexpected big spikes in demand that would require elasticity, such as rapid service scaling. Predictable jobs do not require the immediate deployment of extra resources to respond to unexpected traffic to your cluster. You do not need to shut down resources when traffic drops suddenly.

Related Information

Apache Tez execution engine for Apache Hive

Migration to Cloudera Data Warehouse

If your Hive jobs are unpredictable, consider migrating to Cloudera Data Warehouse (CDW) and running your Hive workloads on LLAP mode. LLAP offers ETL performance and scalability for complex data warehousing jobs.

Such a move can meet or exceed your customer satisfaction requirements, or cut the expense of running your jobs, or both. You set up automatic scaling up of compute resources when needed, or shutting down when not needed.

LLAP along with the query isolation feature is best suited for data-intensive queries, such as ETL queries, and require auto-scaling based on the total scan size of the query.

If your Hive jobs are unpredictable and if you are running complex interactive queries, migrate to CDW. You perform the following configuration when taking this path:

- Configure CDW according to your plan for scaling and concurrency
- Tune the Hive Virtual Warehouse to handle peak workloads

When you create a CDW Virtual Warehouse, you configure the size and concurrency of queries to set up LLAP. The configuration is simple compared to HDP configurations of LLAP. In Hive Virtual Warehouses, each size setting indicates the number of concurrent queries that can be run. For example, an X-Small Hive on LLAP Virtual Warehouse can run 2 TB of data in its cache.

Related Information

Apache Tez execution engine for Apache Hive
Query isolation for data-intensive queries
Auto scaling of Virtual Warehouses in CDW
Virtual Warehouse sizing requirements
Tuning Virtual Warehouses

Apache Tez processing of Hive jobs

After migrating to CDP Private Cloud Base CDP Public Cloud, or Cloudera Data Warehouse (CDW), you must understand how the Apache Tez execution engine is used to run your Hive workloads.

Apache Tez provides the framework to run a job that creates a graph with vertices and tasks. The entire execution plan is created under this framework. Apache Tez provides the following execution modes:

- Container mode — Every time you run a Hive query, Tez requests a container from YARN.
- LLAP mode — Every time you run a Hive query, Tez asks the LLAP daemon for a free thread, and starts running a fragment.

SQL syntax in Hive is the same irrespective of execution mode used in Hive. In CDP Private Cloud Base and CDP Public Cloud, Tez always runs in container mode and is commonly called Hive on Tez. In CDW, Tez always runs in LLAP mode. You can use the query isolation feature if you are running complex ETL workloads.

Migration paths for Spark users

If you are on HDP and executing Spark workloads in Hive LLAP mode, and you want to upgrade to CDP, you can follow the migration path that matches your security needs. It is recommended that you upgrade to CDP Private Cloud Base and choose either Hive Warehouse Connector (HWC) or native Spark readers to query Hive from Spark.

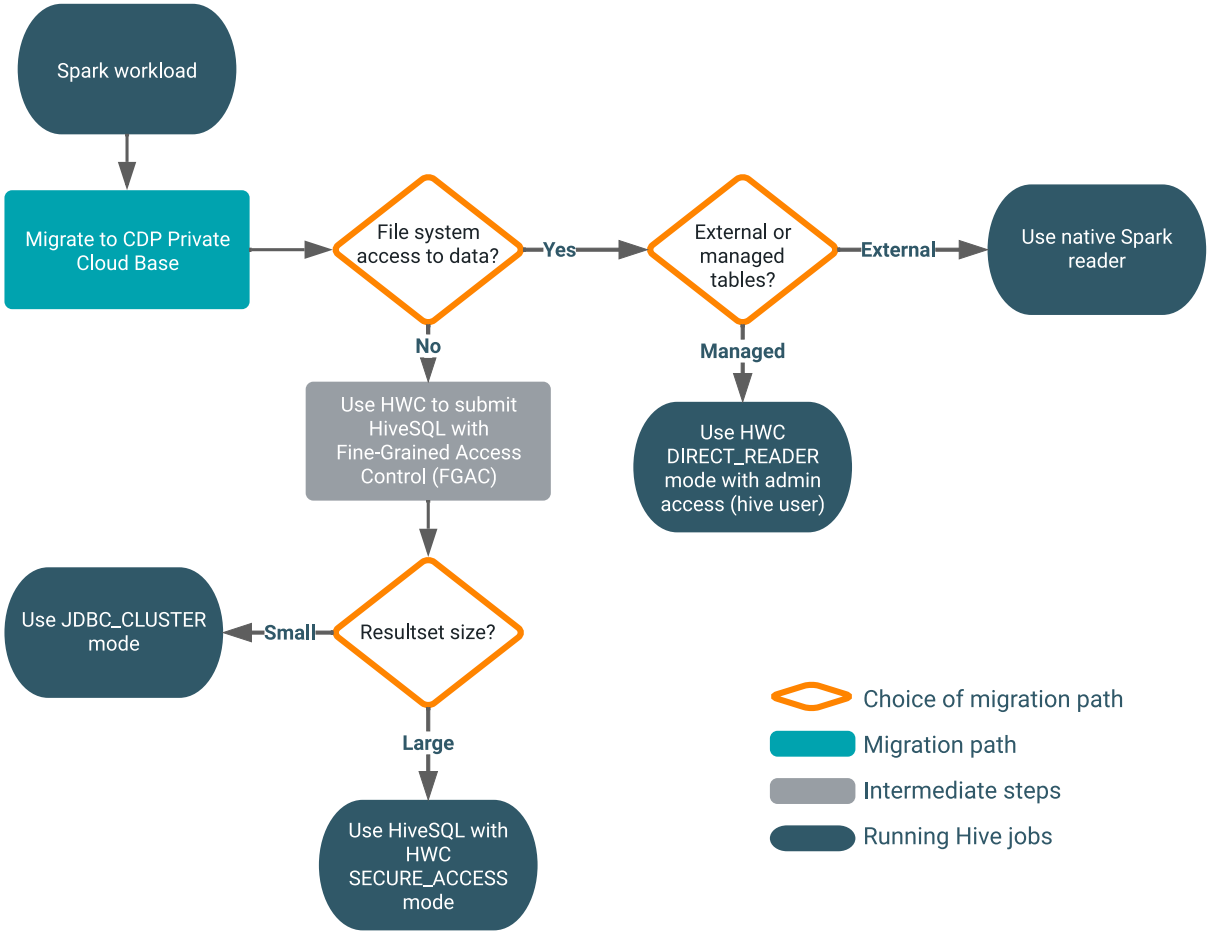
As a replacement for the HWC LLAP execution mode in HDP, you can use the HWC Secure Access Mode in CDP Private Cloud Base that offers fine-grained access control (FGAC) column masking and row filtering to secure managed (ACID), or even external, Hive table data that you read from Spark.

The following migration paths are recommended based on certain factors:

Migrate to CDP Private Cloud Base

- Use HWC JDBC Cluster mode — If the user does not have access to data in the file system and if the database query returns are less than 1 GB of data.
- Use HWC Secure access mode — If the user does not have access to data in the file system and if the database query returns are more than 1 GB of data.
- Use HWC Direct reader mode — If the user has access to data in the file system and if you are querying Hive managed tables.
- Use native Spark reader — If the user has access to data in the file system and if you are querying Hive external tables.

The following diagram shows these recommended paths:



Migration to CDP Private Cloud Base

If you are a Spark user migrating from HDP to CDP and accessing Hive workloads through the Hive Warehouse Connector (HWC), consider migrating to CDP Private Cloud Base and based on your use cases, use the various HWC read modes to access Hive managed tables from Spark.

Use HWC JDBC Cluster mode

If the user does not have access to the file systems (restricted access), you can use HWC to submit HiveSQL from Spark with benefits of fine-grained access control (FGAC), row filtering and column masking, to securely access Hive tables from Spark.

However, if the size of your database query returns are less than 1 GB of data, it is recommended that you use HWC JDBC Cluster mode in which Spark executors connect to Hive through JDBC, and execute the query. Larger workloads are not recommended for JDBC reads in production due to slow performance.

Use HWC Secure access mode

If the user does not have access to the file systems (restricted access) and if the size of database query returns are greater than 1 GB of data, it is recommended to use HWC Secure access mode that offers fine-grained access control (FGAC), row filtering and column masking to access Hive table data from Spark.

Secure access mode enables you to set up an HDFS staging location to temporarily store Hive files that users need to read from Spark and secure the data using Ranger FGAC.

Use HWC Direct reader mode

If the user has access to the file systems (ETL jobs do not require authorization and run as super user) and if you are accessing Hive managed tables, you can use the HWC Direct reader mode to allow Spark to read directly from the managed table location.

Important: This workload must be run with 'hive' user permissions.

If you are querying Hive external tables, use Spark native readers to read the external tables from Spark.

Related Information

Row-level filtering and column masking in Hive

Introduction to HWC JDBC read mode

Introduction to HWC Secure access mode

Introduction to HWC Direct reader mode

HWC changes from HDP to CDP

You need to understand the Hive Warehouse Connector (HWC) changes from HDP to CDP. Extensive HWC documentation can prepare you to update your HWC code to run on CDP. In CDP, methods and the configuration of the HWC connections differ from HDP.

Deprecated methods

The following methods have been deprecated in Cloudera Runtime 7.1.7:

- hive.execute()
- hive.executeQuery()

The HWC interface is simplified in CDP resulting in the convergence of the execute/ executeQuery methods to the sql method. The execute/ executeQuery methods are deprecated and will be removed from CDP in a future release. Historical calls to execute/ executeQuery used the JDBC connection and were limited to 1000 records. The 1000 record limitation does not apply to the sqlmethod, although using JDBC cluster mode is recommended only for production for workloads having a data size of 1GB or less. Larger workloads are not recommended for JDBC reads in production due to slow performance.

Although the old methods are still supported in CDP for backward compatibility, refactoring your code to use the sql method for all configurations (JDBC client, Direct Reader V1 or V2, and Secure Access modes) is highly recommended.

Recommended method refactoring

The following table shows the recommended method refactoring:

API	From HDP	To CDP	HDP Example	CDP Example
HWC sql API	execute and executeQuery methods	sql method	hive.execute("select * from default.hwctest").show(1, false)	hive.sql("select * from default.hwctest").show(1, false)
Spark sql API	sql and spark.read.table methods	No change	sql("select * from managedTable").show scala> spark.read.table("managedTable").show	sql("select * from managedTable").show scala> spark.read.table("managedTable").show
DataFrames API	spark.read.format method	No change	val df = spark.read.format("HiveAcid").options(Map("tabl	val df = spark.read.format("HiveAcid").options(Map("tabl

API	From HDP	To CDP	HDP Example	CDP Example
			e" -> "default.acidtbl")).load()	e" -> "default.acidtbl")).load()

Deprecated and changed configurations

HWC read configuration is simplified in CDP. You use a common configuration for Spark Direct Reader, JDBC Cluster, or Secure Access mode.

The following HWC configurations that you might have used in HDP 3.1.5 cannot be used in CDP:

- --conf spark.hadoop.hive.llap.daemon.service.hosts
- --conf spark.hadoop.hive.zookeeper.quorum

Recommended configuration refactoring

Refactor configuration code to remove unsupported configurations. Use the following common configuration property: spark.datasource.hive.warehouse.read.mode.

You can transparently read data from Spark with HWC in different modes using just spark.sql("<query>").

Secured cluster configurations

In secured cluster configurations, you must set configurations as described in the HWC documentation for CDP:

- --conf "spark.security.credentials.hiveserver2.enabled=true"
- --conf "spark.sql.hive.hiveserver2.jdbc.url.principal=hive/_HOST@ROOT.HWX.SITE"

  The jdbc url must not contain the jdbc url principal and must be passed as shown here.

Deprecated features

The following features have been deprecated and will be removed from CDP in a future release:

- Catalog browsing
- JDBC client mode configuration

Related Information

Introduction to Hive Warehouse Connector