Hortonworks Data Platform 2.0

Data Integration Services with HDP

(Mar 18, 2013)

docs.hortonworks.com

Hortonworks Data Platform 2.0: Data Integration Services with HDP

The Hortonworks Data Platform, powered by Apache Hadoop, is a massively scalable and 100% open source platform for storing, processing and analyzing large volumes of data. It is designed to deal with data from many sources and formats in a very quick, easy and cost-effective manner. The Hortonworks Data Platform consists of the essential set of Apache Hadoop projects including YARN, Hadoop Distributed File System (HDFS), HCatalog, Pig, Hive, HBase, ZooKeeper and Ambari. Hortonworks is the major contributor of code and patches to many of these projects. These projects have been integrated and tested as part of the Hortonworks Data Platform release process and installation and configuration tools have also been included.

Unlike other providers of platforms built using Apache Hadoop, Hortonworks contributes 100% of our code back to the Apache Software Foundation. The Hortonworks Data Platform is Apache-licensed and completely open source. We sell only expert technical support, training and partner-enablement services. All of our technology is, and will remain free and open source.

Please visit the Hortonworks Data Platform page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the Support or Training page. Feel free to Contact Us directly to discuss your specific needs.

Table of Contents

1. Using Apache Hive

Hortonworks Data Platform deploys Apache Hive for your Hadoop cluster.

Hive is a data warehouse infrastructure built on top of Hadoop. It provides tools to enable easy data ETL, a mechanism to put structures on the data, and the capability for querying and analysis of large data sets stored in Hadoop files.

Hive defines a simple SQL-like query language, called QL, that enables users familiar with SQL to query the data. At the same time, this language also allows programmers who are familiar with the MapReduce framework to be able to plug in their custom mappers and reducers to perform more sophisticated analysis that may not be supported by the built-in capabilities of the language.

In this document:

- Hive Documentation
- Using Tez Service with Hive
- Hive JIRAs

1.1. Hive Documentation

Documentation for Hive can be found in wiki docs and javadocs.

1. The Hive wiki is organized in four major sections:
   - General Information about Hive
     - Getting Started
     - Presentations and Papers about Hive
     - Hive Mailing Lists
   - User Documentation
     - Hive Tutorial
     - HiveQL Language Manual – **new information** for the `CREATE TABLE` statement:

       ```
       ... [STORED AS file_format] ...
       file_format:
       : SEQUENCEFILE
       | TEXTFILE
       | RCFILE
       | ORC
       | INPUTFORMAT input_format_classname OUTPUTFORMAT
        output_format_classname
       ```

       and the `hive.default.fileformat` variable's description:

"Default file format for CREATE TABLE statement. Options are TextFile, SequenceFile, **RCFile, and Orc**."

  - Hive Operators and Functions
  - Hive Web Interface
  - Hive Client
  - Avro SerDe

- Administrator Documentation
  - Installing Hive
  - Configuring Hive – **new information** for the `hive.default.fileformat` variable's description:

    "Default file format for CREATE TABLE statement. Options are TextFile, SequenceFile, RCFile, **and Orc**."

  - Setting Up the Metastore
  - Setting Up Hive Web Interface
  - Setting Up Hive Server
  - Hive on Amazon Web Services
  - Hive on Amazon Elastic MapReduce

- Resources for Contributors
  - Hive Developer FAQ
  - How to Contribute
  - Hive Developer Guide
  - Plugin Developer Kit
  - Unit Test Parallel Execution
  - Hive Architecture Overview
  - Hive Design Docs
  - Full-Text Search over All Hive Resources
  - Project Bylaws

2. Supplementary documentation describes new features, including:

- ORC file format

- HiveServer2 JDBC
- Decimal data type
- Windowing and analytics functions
- Optimized joins
- Metastore server security (authorization and authentication)

3. Javadocs describe the Hive API. The supplementary documentation includes a complete set of Javadocs for this release, including the ORC file format.
4. Hive indexing was added in version 0.7.0; documentation and examples can be found here:
    - Indexes – design document (lists the indexing JIRAs with current status, starting with HIVE-417)
    - Create/Drop Index – HiveQL language manual
    - Bitmap indexes – added in Hive version 0.8.0 (Jira HIVE-1803)
    - Indexed Hive – overview and examples by Prafulla Tekawade and Nikhil Deshpande, October 2010
    - Tutorial: SQL-like join and index with MapReduce using Hadoop and Hive – blog by Ashish Garg, April 2012

1.2. Using Tez Service with Hive

Tez is the next generation Hadoop Query Processing framework written on top of YARN.

Tez AM is a new and improved implementation of the MapReduce application that supports container reuse. This allows jobs to run faster on clusters that have limited resources per job. On smaller clusters, it reduces the time for a job to finish by efficiently using a container to run more than one task.

The **Tez AMPoolService** or **Tez Service** is a service that launches and makes available a pool of pre-launched MapReduce AMs (Tez AMs). These AMs in the pool can, in turn, be configured to pre-allocate a number of containers to allow jobs to be launched and completed faster. To use the Tez Service, the clients must submit the jobs to this service instead of the ResourceManager.

Use the instructions provided here to submit Hive queries to Tez Service.

1.3. Hive JIRAs

Issue tracking for Hive bugs and improvements can be found here: Hive JIRAs.

2. Using HDP for Metadata Services (HCatalog)

Hortonworks Data Platform deploys Apache HCatalog to manage the metadata services for your Hadoop cluster.

Apache HCatalog is a table and storage management service for data created using Apache Hadoop. This includes:

- Providing a shared schema and data type mechanism.
- Providing a table abstraction so that users need not be concerned with where or how their data is stored.
- Providing interoperability across data processing tools such as Pig, MapReduce, and Hive.

Start the HCatalog CLI with the command '`<hadoop-install-dir>\hcatalog-0.5.0\bin\hcat.cmd`'.

HCatalog Documentation

For details about HCatalog see the Apache HCatalog documentation, which includes the following resources:

- HCatalog Overview
- Installation from Tarball – see below for corrections based on HCATALOG-625.
- Load and Store Interfaces
- Input and Output Interfaces
- Reader and Writer Interfaces
- Command Line Interface
- Storage Formats
- Dynamic Partitioning
- Notification
- Authorization
- API Documentation

Using WebHCat (Templeton)

WebHCat provides a REST-like web API for HCatalog and related Hadoop components.

Note

WebHCat was originally named *Templeton*, and both terms may still be used interchangeably.

For details about WebHCat, see the following resources:

- Overview
- Installation
- Configuration
- **Reference**
  - Resource List
  - :version
  - status
  - version
  - ddl
  - mapreduce/streaming
  - mapreduce/jar
  - pig
  - hive
  - queue
  - queue/:jobid (GET)
  - queue/:jobid (DELETE)
- API Docs

Corrections to Installation from Tarball (see HCATALOG-625)

- *Replace the section "Building a tarball" with this:*

  If you downloaded HCatalog from Apache or another site as a source release, you will need to first build a tarball to install. You can tell if you have a source release by looking at the name of the object you downloaded. If it is named `hcatalog-src-0.5.0-incubating.tar.gzs` (notice the **src** in the name) then you have a source release.

  If you do not already have Apache Ant installed on your machine, you will need to obtain it. You can get it from the Apache Ant website. Once you download it, you will need to unpack it somewhere on your machine. The directory where you unpack it will be referred to as *ant_home* in this document.

  To produce a binary tarball from downloaded src tarball, execute the following steps:

  ```
  tar xzf hcatalog-src-0.5.0-incubating.tar.gz
  cd hcatalog-src-0.5.0-incubating
  ant_home/bin/ant package
  ```

The tarball for installation should now be at `build/hcatalog-0.5.0-incubating.tar.gz` .

- *In the "Thrift Server Setup" section:*
  - *In the third paragraph, replace* **Hive 0.9** *with* **the current version of Hive**.
  - *Replace these commands:*

    ```
    tar zxf hcatalog-0.5.0.tar.gz
    cd hcatalog-0.5.0
    ```

    with these:

    ```
    tar zxf hcatalog-0.5.0-incubating.tar.gz
    cd hcatalog-0.5.0-incubating
    ```

  - *In the next paragraph, add this sentence:*

    If there is no `hive-site.xml` file in the hive conf directory, copy *hcat_home*`/etc/hcatalog/proto-hive-site.xml` and rename it `hive-site.xml` in *hive_home*`/conf/`.

- *In the "Client Installation" section, replace this command:*

  ```
  tar zxf hcatalog-0.5.0-incubating.tar.gz
  ```

  with this:

  ```
  tar zxf hcatalog-0.5.0-incubating.tar.gz
  ```

Additional Information

For more details on the Apache HCatalog project, use the following resources:

- HCatalog Wiki
- HCatalog Mailing Lists