Hortonworks Data Platform

Data Integration Services with HDP

(Jul 2, 2013)

docs.hortonworks.com

Hortonworks Data Platform: 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 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:

1.1. Hive Documentation

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

1. Javadocs describe the Hive API.
2. 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
     - Hive Operators and Functions
     - Hive Web Interface
     - Hive Client

  - HiveServer2 Client
  - Avro SerDe
- Administrator Documentation
  - Installing Hive
  - Configuring Hive
  - 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

1.2. New Feature: Vectorization

Vectorization allows Hive to process a batch of rows together instead of processing one row at a time. Each batch consists of a *column vector* which is usually an array of primitive types. Operations are performed on the entire column vector, which improves the instruction pipelines and cache usage.

1.2.1. Enable Vectorization in Hive

To enable vectorization, set this configuration parameter:

- `hive.vectorized.execution.enabled=true`

When vectorization is enabled, Hive examines the query and the data to determine whether vectorization can be supported. If it cannot be supported, Hive will execute the query with vectorization turned off.

1.2.2. Log Information about Vectorized Execution of Queries

The Hive client will log, at the `info` level, whether a query's execution is being vectorized. More detailed logs are printed at the `debug` level.

The client logs can also be configured to show up on the console.

1.2.3. Supported Functionality

The current implementation supports only single table read-only queries. DDL queries or DML queries are not supported.

The supported operators are selection, filter and group by.

Partitioned tables are supported.

These data types are supported:

- tinyint
- smallint
- int
- bigint
- boolean
- float
- double
- timestamp
- string

These expressions are supported:

- Comparison: >, >=, <, <= , =, !=
- Arithmetic: plus, minus, multiply, divide, modulo
- Logical: AND, OR
- Aggregates: sum, avg, count, min, max

Only the ORC file format is supported in the current implementation.

1.2.4. Unsupported Functionality

All datatypes, file formats, and functionality *not listed in the previous section* are currently unsupported.

Two unsupported features of particular interest are the logical expression NOT and the `cast` operator. For example, a query such as `select x,y from T where a = b` will not vectorize if `a` is integer and `b` is double. Although both int and double are supported, casting of one to another is not supported.

1.3. Hive ODBC Driver

Hortonworks provides a Hive ODBC driver that allows you to connect popular Business Intelligence (BI) tools to query, analyze and visualize data stored within the Hortonworks Data Platform.

- Download the Hortonworks Hive ODBC driver from here.
- The instructions on installing and using this driver are available here.

1.4. 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.

1.5. 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`'.

Note

HCatalog 0.5.0 was the final version released from the Apache Incubator. In March 2013, HCatalog graduated from the Apache Incubator and became part of the Apache Hive project. New releases of Hive include HCatalog, starting with Hive 0.11.0.

HCatalog includes two documentation sets:

1. General information about HCatalog

   This documentation covers installation and user features. The next section, Using HCatalog, provides links to individual documents in the HCatalog documentation set.

2. WebHCat information

   WebHCat is a web API for HCatalog and related Hadoop components. The section Using WebHCat provides links to user and reference documents, and includes a technical update about standard WebHCat parameters.

For more details on the Apache Hive project, including HCatalog and WebHCat, see Using Apache Hive and the following resources:

- Hive project home page
- Hive wiki home page
- Hive mailing lists

2.1. Using HCatalog

For details about HCatalog, see the following resources in the HCatalog documentation set:

- HCatalog Overview
- Installation From Tarball
- Load and Store Interfaces
- Input and Output Interfaces
- Reader and Writer Interfaces
- Command Line Interface
- Storage Formats
- Dynamic Partitioning
- Notification
- Storage Based Authorization

2.2. Using WebHCat

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

Note

WebHCat was originally named *Templeton*, and both terms may still be used interchangeably. For backward compatibility the Templeton name still appears in URLs, log file names, etc.

For details about WebHCat (Templeton), see the following resources:

- Overview
- Installation
- Configuration
- **Reference**
  - Resource List
  - GET :version
  - GET status
  - GET version
  - DDL Resources: Summary and Commands
  - POST mapreduce/streaming
  - POST mapreduce/jar
  - POST pig

- POST hive
- GET queue
- GET queue/:jobid
- DELETE queue/:jobid

2.2.1. Technical Update: WebHCat Standard Parameters

The "Security" section of the WebHCat Overview should be updated with information in the Note below:

2.2.1.1. Security

The current version supports two types of security:

- Default security (without additional authentication)
- Authentication via Kerberos

2.2.1.1.1. Standard Parameters

Every REST resource can accept the following parameters to aid in authentication:

- user.name: The user name as a string. Only valid when using default security.
- SPNEGO credentials: When running with Kerberos authentication.

Note

The user.name parameter is part of POST parameters for POST calls, and part of the URL for other calls.

For example, to specify user.name in a GET :table command:

```
% curl -s 'http://localhost:50111/templeton/v1/ddl/database/default/
table/my_table?user.name=ctdean'
```

And to specify user.name in a POST :table command:

```
% curl -s -d user.name=ctdean \
       -d rename=test_table_2 \
       'http://localhost:50111/templeton/v1/ddl/database/default/
table/test_table'
```

2.2.1.1.2. Security Error Response

If the user.name parameter is not supplied when required, the following error will be returned:

```
{
  "error": "No user found.  Missing user.name parameter."
}
```

3. Using HDP for Workflow and Scheduling (Oozie)

Hortonworks Data Platform deploys Apache Oozie for your Hadoop cluster.

Oozie is a server-based workflow engine specialized in running workflow jobs with actions that execute Hadoop jobs, such as MapReduce, Pig, Hive, Sqoop, HDFS operations, and sub-workflows. Oozie supports coordinator jobs, which are sequences of workflow jobs that are created at a given frequency and start when all of the required input data is available. A command-line client and a browser interface allow you to manage and administer Oozie jobs locally or remotely.

For additional Oozie documentation, use the following resources:

- Quick Start Guide
- Developer Documentation
  - Oozie Workflow Overview
  - Running the Examples
  - Workflow Functional Specification
  - Coordinator Functional Specification
  - Bundle Functional Specification
  - EL Expression Language Quick Reference
  - Command Line Tool
  - Workflow Rerun
  - Email Action
  - Writing a Custom Action Executor
  - Oozie Client Javadocs
  - Oozie Core Javadocs
  - Oozie Web Services API
- Administrator Documentation
  - Oozie Installation and Configuration
  - Oozie Monitoring
  - Command Line Tool