# **Apache Hive**

The [**Apache Hive™**](http://hive.apache.org/) data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage and queried using SQL syntax.

Built on top of [**Apache Hadoop™**](http://hadoop.apache.org/), Hive provides the following features:

* Tools to enable easy access to data via SQL, thus enabling data warehousing tasks such as extract/transform/load (ETL), reporting, and data analysis.
* A mechanism to impose structure on a variety of data formats
* Access to files stored either directly in [**Apache HDFS**](http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html)[**™**](http://hadoop.apache.org/) or in other data storage systems such as [**Apache HBase**](http://hbase.apache.org/)[**™**](http://hadoop.apache.org/)
* Query execution via [Apache Tez](http://tez.apache.org/)[**™**](http://hadoop.apache.org/), [Apache Spark](http://spark.apache.org/)[**™**](http://hadoop.apache.org/), or [MapReduce](http://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html)
* Procedural language with HPL-SQL
* Sub-second query retrieval via [Hive LLAP](https://cwiki.apache.org/confluence/display/Hive/LLAP), [Apache YARN](https://hadoop.apache.org/docs/r2.7.2/hadoop-yarn/hadoop-yarn-site/YARN.html) and [Apache Slider](https://slider.incubator.apache.org/).

Hive is not designed for online transaction processing (OLTP) workloads. It is best used for traditional data warehousing tasks.

Hive is designed to maximize scalability (scale out with more machines added dynamically to the Hadoop cluster), performance, extensibility, fault-tolerance, and loose-coupling with its input formats.

—----------------------------------------------------------------------------------------------------------------------------

**INSTALLATION OF HIVE**

<https://dlcdn.apache.org/hive/>

## **Installation and Configuration**

You can install a stable release of Hive by downloading a tarball, or you can download the source code and build Hive from that.

#### **Running HiveServer2 and Beeline**

### **Requirements**

* Java 1.7  
  *Note:* Hive versions [1.2](https://issues.apache.org/jira/browse/HIVE/fixforversion/12329345/?selectedTab=com.atlassian.jira.jira-projects-plugin:version-summary-panel) onward require Java 1.7 or newer. Hive versions 0.14 to 1.1 work with Java 1.6 as well. Users are strongly advised to start moving to Java 1.8 (see [HIVE-8607](https://issues.apache.org/jira/browse/HIVE-8607)).
* Hadoop 2.x (preferred), 1.x (not supported by Hive 2.0.0 onward).  
  Hive versions up to 0.13 also supported Hadoop 0.20.x, 0.23.x.
* Hive is commonly used in production Linux and Windows environment. Mac is a commonly used development environment. The instructions in this document are applicable to Linux and Mac. Using it on Windows would require slightly different steps.

### **Installing Hive from a Stable Release**

Start by downloading the most recent stable release of Hive from one of the Apache download mirrors (see [Hive Releases](https://hive.apache.org/downloads.html)).

Next you need to unpack the tarball. This will result in the creation of a subdirectory named hive-x.y.z (where x.y.z is the release number):

$ tar -xzvf hive-x.y.z.tar.gz

Set the environment variable HIVE\_HOME to point to the installation directory:

$ cd hive-x.y.z

$ export HIVE\_HOME={{pwd}}

Finally, add $HIVE\_HOME/bin to your PATH:

$ export PATH=$HIVE\_HOME/bin:$PATH

### **Building Hive from Source**

The Hive GIT repository for the most recent Hive code is located here: git clone <https://git-wip-us.apache.org/repos/asf/hive.git> (the master branch).

All release versions are in branches named "branch-0.#" or "branch-1.#" or the upcoming "branch-2.#", with the exception of release 0.8.1 which is in "branch-0.8-r2". Any branches with other names are feature branches for works-in-progress. See [Understanding Hive Branches](https://cwiki.apache.org/confluence/display/Hive/HowToContribute#HowToContribute-UnderstandingHiveBranches) for details.

As of 0.13, Hive is built using [Apache Maven](http://maven.apache.org/).

#### **Compile Hive on master**

To build the current Hive code from the master branch:

$ git clone https://git-wip-us.apache.org/repos/asf/hive.git

$ cd hive

$ mvn clean package -Pdist [-DskipTests -Dmaven.javadoc.skip=true]

$ cd packaging/target/apache-hive-{version}-SNAPSHOT-bin/apache-hive-{version}-SNAPSHOT-bin

$ ls

LICENSE

NOTICE

README.txt

RELEASE\_NOTES.txt

bin/ (all the shell scripts)

lib/ (required jar files)

conf/ (configuration files)

examples/ (sample input and query files)

hcatalog / (hcatalog installation)

scripts / (upgrade scripts for hive-metastore)

—-------------------------------------------------------------------------

## Hive Query Language

HiveQL is an SQL-like query language for Hive. It mostly mimics SQL syntax for creation of tables, loading data into tables and querying the tables. HiveQL also allows users to embed their custom map-reduce scripts. These scripts can be written in any language using a simple row-based streaming interface – read rows from standard input and write out rows to standard output. This flexibility comes at a cost of a performance hit caused by converting rows from and to strings. However, we have seen that users do not mind this given that they can implement their scripts in the language of their choice. Another feature unique to HiveQL is multi-table insert. In this construct, users can perform multiple queries on the same input data using a single HiveQL query. Hive optimizes these queries to share the scan of the input data, thus increasing the throughput of these queries several orders of magnitude. We omit more details due to lack of space. For a more complete description of the HiveQL language see the [language manual](https://cwiki.apache.org/confluence/display/Hive/LanguageManual).

