Hive Installation Guide

A guide to install to Hive

Version 2.2

Hive Installation Guide

Table of Contents

Pre-requisites:

- → JDK 1.6
- → Maven (Even ANT can be used but for this exercise we will use maven)
- → Subversion
- → Database Engine (Mysql, Postgres, Oracle or Derby)

Basic Information:

- → HIVE is a tool to enable ETL
- → Gives you a SQL like language known as Hive Query Language (HQL).
- → The HQL can be used for querying data residing on HDFS/HBASE.
- → Depending upon the type of query issued, map-reduce would be called.
- → Allow you to inject your own functions using UDF's user defined functions.
- → Allows you to work on multiple formats using FileFormats, SerDe (Serialization Deserailization etc.)
- → Hive 0.11 + comes with two inbuilt components i.e. HCatalog and WebHCat.
- → Hcatalog: A table and storage management layer on top of hadoop which allows users to read/write data easily with different processing tools like PIG/Mapreduce. The overhead of knowing beforehand what format the file is in, which compression format is used, location of the files is taken away from the developer.
- → WebHCat: Same as HCatalog but allows REST API calls to allow read/write of data using YARN, PIG, Graph technologies etc.
- → Hive has three kinds of metastores available for storing table information:

1. **Embedded Metastore:**

   Embedded Derby database, default installation. Writes all the table related information on the disk and the path corresponds to ./ i.e. From whereever the hive binary is called. Good for testing purpose, Single user session, each hive client will see tables owned or created by him.

2. **Local Metastore:**

   Uses one of the database engine to store all table information. Multiple hive clients can use and access tables. All clients can see all tables. Good for Production.

3. **Remote Metastore:**

   In remote metastore setup, all Hive Clients will make a connection to a metastore server which in turn queries the datastore for metadata. Metastore server and client communicate using Thrift Protocol. Starting with Hive 0.5.0, you can start a Thrift server by executing the following command:

   ```
   hive --service metastore
   ```

Further reading:

https://cwiki.apache.org/confluence/display/Hive/AdminManual+MetastoreAdmin#AdminManualMetastoreAdmin-RemoteMetastore

Steps to Install:

1. Check out code from svn in /tmp location

   svn co http://svn.apache.org/repos/asf/hive/trunk hive

2. ***cd /tmp/hive***

3. ***mvn clean install -DskipTests -Phadoop-1,dist***

   The abive command starts building Hive against Hadoop-1, if you wish to build it against hadoop-2 use ***-Phadoop-2,dist***

4. Once the build process completes you will have a hive binary distribution tarball available in your target directory. Move that tar ball to the $HIVE_INSTALL directory.

   $HIVE_INSTALL = the directory where you want to install hive

5. Extract the tar ball.

6. Create a soft link to the extracted directory.

7. Set HIVE_HOME in either .bashrc or /etc/profile file and also make sure HIVE binaries are available in the PATH variable, reload the file using source command.

   export HIVE_HOME=$HIVE_INSTALL

   export PATH=$PATH:$HIVE_HOME/bin

   source .bashrc

8. Create a /tmp directory on HDFS if not created already and change permission to 775

9. By default HIVE assumes it will be run using the hive user thus requires you to create a warehouse directory in the hive user's home directory on the HDFS with 775 permission

```
hadoop fs -mkdir /tmp /user/hive/warehouse

hadoop fs -chmod 775 /tmp

hadoop fs -chmod 775 /user/hive/warehouse
```

If you wish to use a different location for storing Hive related table information on the HDFS make sure you modify the hive.metastore.warehouse.dir property in hive-site.xml the default value is /user/hive/warehouse

10. Create a database and username-password in your loved Database Engine, and update the hive-site.xml with proper values. Below are the properties that need to be updated:

```
<property>
        <name>hive.metastore.local</name>
        <value>true</value>
</property>
<property>
        <name>javax.jdo.option.ConnectionURL</name>
        <value>jdbc:postgresql://localhost:5432/hivedb</value>
</property>
<property>
         <name>javax.jdo.option.ConnectionDriverName</name>
         <value>org.postgresql.Driver</value>
</property>
<property>
       <name>javax.jdo.option.ConnectionUserName</name>
       <value>hiveuser</value>
</property>
<property>
         <name>javax.jdo.option.ConnectionPassword</name>
         <value>hiveuser</value>
</property>
```

The above property sets up postgres as the database engine with hivedb as the database. Setting these properties will enable you to create a local metastore for hive.

11. Once the database is created and the properties file updated, you will have to create a schema in the database.

    [If using mysql, provide?createDatabaseIfNotExist=true in connectionURL which would create the DB for you].

    The schema files can be found at ***$HIVE_HOME/scripts/metastore/upgrade***

    The above location will have folders for different db engines. Select the one you are using, inside it you will find a hive-schema-$version.sql file, where $version is the hive version available till now, choose the file which corresponds to the version you've built and execute that in your DB, which would create the hive required tables.

12. Also make sure that before using HIVE, you need to make sure the appropriate database connector is available in the ***$HIVE_HOME/lib*** directory.

13. Start using Hive CLI.