Apache Hive and HDFS Usage

This part number 2 of our Data Analysis series of labs/experiments. In this lab we will download, install and configure Apache Hive so we can use it to run our analysis over Hadoop.

From the Apache Hive developers themselves:
`“”” The Apache Hive ™ data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL. Structure can be projected onto data already in storage. A command line tool and JDBC driver are provided to connect users to Hive. “”” [1]`

Requirements

- Name Node
- Download the Apache Hive version found here (see download instructions below)
- Add another Network Interface to your Name Node and have it connected to a NAT network. This could be used to access the internet.

System	Hostname	Host-only IP Address	NAT IP Address	Role
VM#1	nnode.yourname.local	172.16.150.10	N/A	Name Node

Note: NAT IP Address will be different for each system, that is why it is listed as Not Available.

Installing Apache Hive

Login to your Name node and then follow the steps below. Please go through them carefully in order to insure you have a clean and working system.

1. Now download Apache Hive inside our 'files' directory
   $ cd files
   $ wget -c
http://us.mirrors.quenda.co/apache/hive/stable-2/apache-hive-2.3.4-bin.tar.gz

2. Extract the files:
   $ tar xvfz apache-hive-2.3.4-bin.tar.gz

3. We are going to move the extracted directory to our user's home directory.
   $ mv apache-hive-2.3.4-bin ../hive

Time to configure our Apache Hive.

4. First, let us create the hive environment configuration file from the available templates. Go to the '**/home/user1/hive/conf**' directory and then do the following:
   $ cp hive-env.sh.templae hive-env.sh

5. Now open the file, go to the end of the file, and make sure you have the following lines at the end:
   export HADOOP_HOME=/home/user1/hadoop
   export HIVE_CONF_DIR=/home/user1/hive/conf

   These lines are to define where to find Hadoop and where to find the configuration files for Apache Hive.

6. More environment variables are needed. Let us open our '.bashrc' file and append the lines below to the end of the file.
   HIVE_HOME=/home/user1/hive
   HIVE_CONF_DIR=/home/user1/hive/conf
   export PATH=/home/user1/hive/bin:$PATH
   export CLASSPATH=$CLASSPATH:/home/user1/hadoop/lib/*:.
   export CLASSPATH=$CLASSPATH:/home/user1/hive/lib/*:.

7. Now within the same directory '**/home/user1/hive/conf**' create a file named 'hive-site.xml'.

8. Open the newly created file and add the settings below to the file.
   ```
   <property>
       <name>javax.jdo.option.ConnectionURL</name>
       <value>jdbc:derby:/home/user1/hive/metastore_db;databaseName=metas
       tore_db;create=true</value>
   </property>
   ```

We need to download a couple of tools and libraries from the Internet, therefore check that you are able to connect to the Internet (using the NATed interface). As a test you can just ping google.com.

9. Now, update the system's repositories with the command below.
   ```
   $ sudo apt update
   ```

10. Assuming all went as we want, now let us install the Apache Derby Database. This could be done using the command below.
    ```
    $ sudo apt install derby-tools libderby-java libderbyclient-java
    ```

11. Let us go to the hive directory to initialize the derby db. First go to the directory:
    ```
    $ cd ~/hive
    ```

12. Then run the initialize command:
    ```
    $ schematool -initSchema -dbType derby
    ```

13. Now, go back to the home directory of the user. This could be done many ways, the easiest is just type "**cd**" and hit Enter.

14. To double check your location within the system, use the "pwd" command:
    ```
    $ pwd
    ```

    You should see:
    ```
    /home/user1
    ```

15. Let us start our hive and see if everything is working. Run the "hive" command:
    ```
    $ hive
    ```

    You should get the hive command prompt as seen below.
    ```
    hive>
    ```

16. Now run the “show tables;” command:
    hive> show tables;

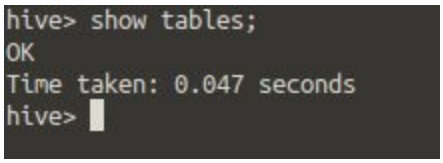
17. You should get a confirmation message similar to the following:
    OK

    Time taken: 5.902 seconds, Fetched: 2 row(s)

18. If you get an error, then exit the hive using:
    $ exit;

19. Now run the following commands:
    $ ls -d $HIVE_HOME/meta*
    $ rm -rf $HIVE_HOME/metastore_db

20. Now go back to step (15) and repeat it again.



Working with Apache Hive

Start your hive and let us do a couple of basic operations within it. One of the really useful website that will help you is the following:
https://www.tutorialspoint.com/sql/index.htm

1. We will be ingesting our movies.csv file into a new database. The movies database has the following basic structure:

   movies(year, imdb, title, budget)

2. In hive let us create our table using the following:

```
CREATE TABLE IF NOT EXISTS movies (year INT, imdb STRING,
title STRING, budget FLOAT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n' ;
```

```
hive> CREATE TABLE IF NOT EXISTS movies (year INT, imdb STRING, title STRING,budget FLOAT)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > LINES TERMINATED BY '\n' ;
OK
Time taken: 1.309 seconds
hive>
```

3. Now download the movies.csv file found here, copy it to your cluster, and then load it into your new '**movies**' table as seen below.

```
LOAD DATA LOCAL INPATH '/home/user1/data/movies.csv'
OVERWRITE INTO TABLE movies;
```

Note: the movies db we used, is a modified version of the original db found here.

```
hive> LOAD DATA LOCAL INPATH '/home/user1/data/movies.csv' OVERWRITE INTO TABLE movies;
Loading data to table default.movies
OK
Time taken: 0.567 seconds
hive>
```

Now you should be able to play with Apache Hive and the data loaded there.