

ROLL NO: 210701268

Exp5: Installation of Hive on Ubuntu

Aim:

To Download and install Hive, Understanding Startup scripts, Configuration files.

Procedure:

1) Step 1: Download and extract it

Download the Apache hive and extract it use tar, the commands given below:

```
$wgethttps://downloads.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz
$ tar -xvf apache-hive-3.1.2-bin.tar.gz
```

2) Step 2: Place different configuration properties in Apache Hive

In this step,do two things:

- Placing Hive Home path in bashrc file: *\$nano .bashrc*
- And append the below lines in it

```
export HIVE_HOME=/home/hadoop/apache-hive-3.1.2-bin
export PATH=$PATH:$HIVE_HOME/bin
export HADOOP_USER_CLASSPATH_FIRST=true
```

- Exporting Hadoop path in Hive-config.sh
- Open the hiveconfig.sh as shown in below

```
$cd apache-hive-3.1.2-bin/bin
```

```
$cp hive-env.sh.template hive-env.sh
```

```
$nano hive-env.sh
```

Append the below commands on it

```
export HADOOP_HOME=/home/Hadoop/Hadoop
```

```
export HIVE_CONF_DIR=/home/Hadoop/apache-hive-3.1.2/conf
```

```
# Set HADOOP_HOME to point to a specific hadoop install directory
# HADOOP_HOME=${bin}/../..//hadoop
export HADOOP_HOME=/home/hadoop/hadoop

# Hive Configuration Directory can be controlled by:
# export HIVE_CONF_DIR=
export HIVE_CONF_DIR=/home/hadoop/apache-hive-3.1.2-bin/conf
# Folder containing extra libraries required for hive compilation/execution can be controlled by:
```

3) Step 3: Install mysql

1. Install mysql in Ubuntu by running this command:

```
$sudo apt update
```

```
$sudo apt install mysql-server
```

2. Alter username and password for MySQLby running below commands:

```
$sudomysql
```

Pops command line interface for MySQL and run the below SQL queries to change username and set password

mysql> SELECT user, host, plugin FROM mysql.user WHERE user = 'root';

```
hadoop@sanjay-VirtualBox:~$ sudo mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 172
Server version: 8.0.34-0ubuntu0.23.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SELECT user, host, plugin FROM mysql.user WHERE user = 'root';
+-----+-----+-----+
| user | host      | plugin                |
+-----+-----+-----+
| root | %         | mysql_native_password |
| root | localhost | auth_socket           |
+-----+-----+-----+
2 rows in set (0.03 sec)

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH 'mysql_native_password' BY 'your_new_password';
Query OK, 0 rows affected (0.04 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)
```

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH 'mysql_native_password' BY 'your_new_password';

mysql> FLUSH PRIVILEGES;

Step 4: Config hive-site.xml

Config the hive-site.xml by appending this xml code and change the username and password according to your MySQL.

\$cd apache-hive-3.1.2-bin/bin

\$cp hive-default.xml.template hive-site.xml

\$nano hive-site.xml

Append these lines into it

Replace root as your username of MySQL

Replace your_new_password as with your password of MySQL

<configuration>

<property>

<name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:mysql://localhost/metastore?createDatabaseIfNotExist=true</value>

</property>

<property>

<name>javax.jdo.option.ConnectionDriverName</name>

<value>com.mysql.cj.jdbc.Driver</value>

</property>

<property>

<name>javax.jdo.option.ConnectionUserName</name>

<value>root</value>

</property>

<property>

```

<name>javax.jdo.option.ConnectionPassword</name>
<value>your_new_password</value>
</property>

<property>
<name>datanucleus.autoCreateSchema</name>
<value>true</value>
</property>

<property>
<name>datanucleus.fixedDatastore</name>
<value>true</value>
</property>

<property>
<name>datanucleus.autoCreateTables</name>
<value>True</value>
</property>

```

```
</configuration>
```

5) Step 5: Setup MySQL java connector:

- First, download the MySQL Connector/J, which is the JDBC driver for MySQL.
- Copy the downloaded MySQL Connector/J JAR file to the Hive library directory. By default, the Hive library directory is usually located at [/path/to/apache-hive-3.1.2/lib/](#) on Ubuntu.
- Use the following command to copy the JAR file:
`$sudo cp /path/to/mysql-connector-java-8.0.15.jar /path/to/apache-hive-3.1.2/lib/` Replace `/path/to/` with the actual path to the JAR file.

6) Step 6: Initialize the Hive Metastore Schema:

Run the following command to initialize the Hive metastore schema:

```
$$HIVE_HOME/bin/schematool -initSchema -dbTypemysql
```

```

hadoop@sanjay-VirtualBox:~$ schematool --dbType mysql --initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:      jdbc:mysql://localhost/metastore?createDatabaseIfNotExist=true
Metastore Connection Driver :  com.mysql.cj.jdbc.Driver
Metastore connection User:    root

```

7) Step 7: Start hive:

- Test Hive by running the Hive shell:
- Copy code hive You should be able to run Hive queries, and metadata will be stored in your MySQL database.

\$hive

```
sudhashreem@sudhashreem-VirtualBox:~/DA/exp4$ cd ..
sudhashreem@sudhashreem-VirtualBox:~/DA$ cd ..
sudhashreem@sudhashreem-VirtualBox:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/sudhashreem/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/sudhashreem/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = f658134c-3a88-4e89-b5c3-a1de3c652802

Logging initialized using configuration in jar:file:/home/sudhashreem/hive/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async: true
Hive Session ID = 88cb5f56-7fc1-44e5-a5f6-2c6a09794be2
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> CREATE DATABASE financials;
OK
Time taken: 0.954 seconds
hive> use financials;
OK
Time taken: 0.185 seconds
hive> CREATE TABLE finance_model( id INT, name STRING);
OK
Time taken: 1.0 seconds
hive> INSERT INTO finance_model VALUES (1,'Alice'),(2,'Bob'),(3,'Candice');
Query ID = sudhashreem_20240917160510_56998f35-1574-4484-9b8f-3ea7049d4d70
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1726567709224_0004, Tracking URL = http://sudhashreem-VirtualBox:8088/proxy/application_1726567709224_0004/
Kill Command = /home/sudhashreem/hadoop/bin/mapred job -kill job 1726567709224 0004
```

Result: Thus, the Apache Hive installation is completed successfully on Ubuntu.

