EX: 05 **REG.NO:210701290**

Installation of Hive on Ubuntu

Aim:

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

Procedure:

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
```

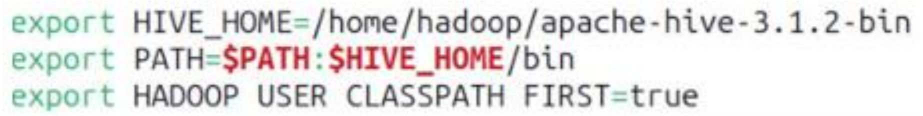
Step 2: Place different configuration properties in Apache Hive

In this step, we are going to 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
```

2. Exporting **Hadoop path in Hive-config.sh** (To communicate with the Hadoop eco system we are defining Hadoop Home path in hive config field) **Open the hive config.sh as shown in below**

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

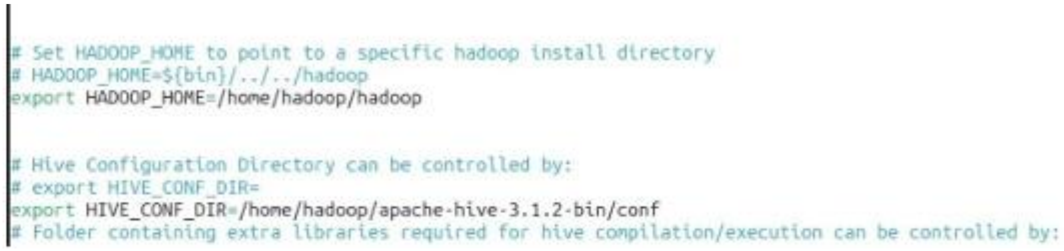
```
$cp hive-env.sh.template hive-env.sh
```

```
$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:
```

Step 3: Install mysql

1. Install mysql in Ubuntu by running this command:

$sudo apt update

$sudo apt install mysql-server

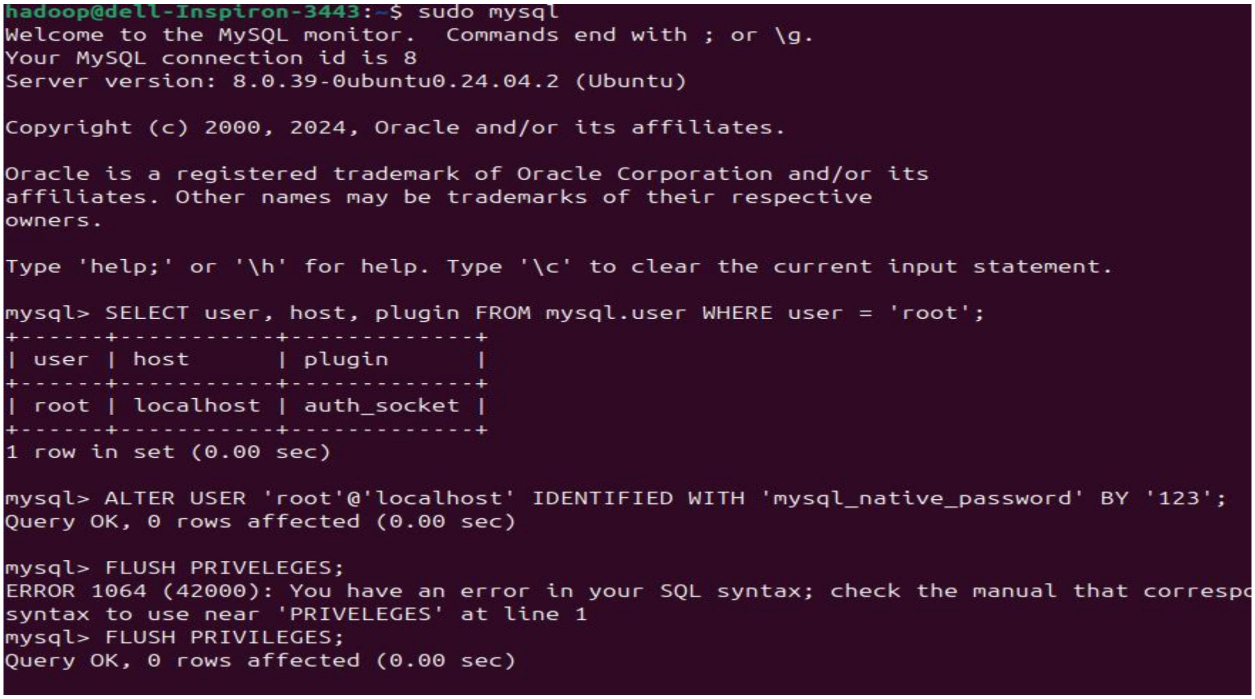
2. Alter username and password for MySQLby running below commands:

$sudomysql

Pops command line interface for MySQLand run the below SQL queries to change username

and set password

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



```
hadoop@dell-Inspiron-3443:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.39-0ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2024, 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 | localhost | auth_socket |
+------+-----------+-------------+
1 row in set (0.00 sec)

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH 'mysql_native_password' BY '123';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVELEGES;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that correspo
syntax to use near 'PRIVELEGES' at line 1
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 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

Replaceyour_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>
```

Step 5: Setup MySQL java connector:

First, you'll need to download the MySQL Connector/J, which is the JDBC driver for MySQL. You can download it from the below link

https://drive.google.com/file/d/1QFhB7Kvcat7a4LzDRe6GcmZva1yAxKz
/view?usp=drive_link

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.

Step 6:Initialize the Hive Metastore Schema:

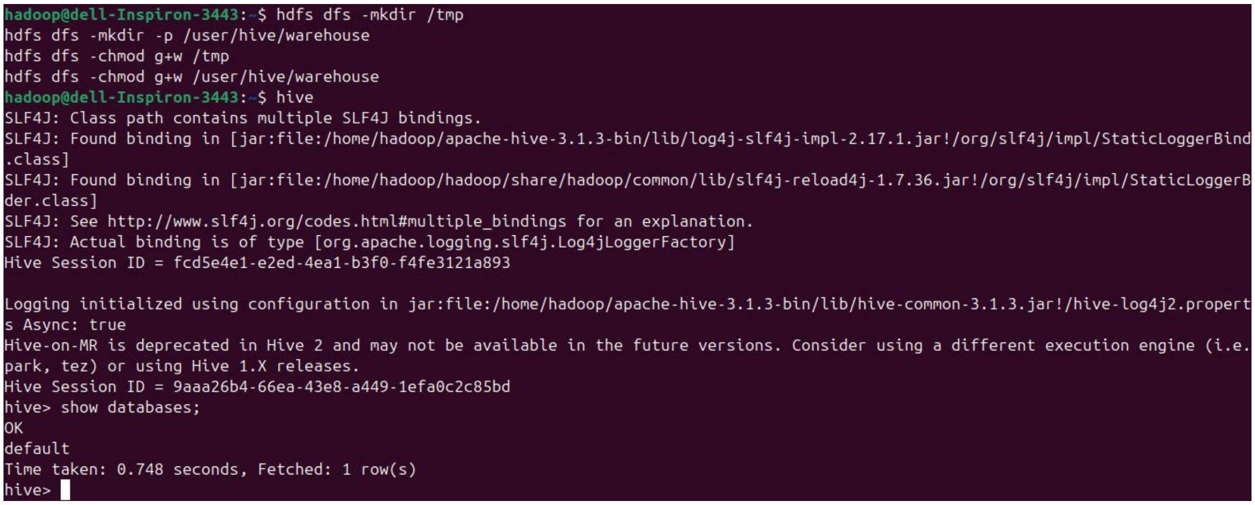
Run the following command to initialize the Hive metastore schema:

$$HIVE_HOME/bin/schematool -initSchema –dbTypemysql

Step 7: Start hive:

You can 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



```
hadoop@dell-Inspiron-3443:~$ hdfs dfs -mkdir /tmp
hdfs dfs -mkdir -p /user/hive/warehouse
hdfs dfs -chmod g+w /tmp
hdfs dfs -chmod g+w /user/hive/warehouse
hadoop@dell-Inspiron-3443:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBind
.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerB
der.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 = fcd5e4e1-e2ed-4ea1-b3f0-f4fe3121a893

Logging initialized using configuration in jar:file:/home/hadoop/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.propert
s Async: true
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.
park, tez) or using Hive 1.X releases.
Hive Session ID = 9aaa26b4-66ea-43e8-a449-1efa0c2c85bd
hive> show databases;
OK
default
Time taken: 0.748 seconds, Fetched: 1 row(s)
hive>
```

Result:

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