

Exp.No.:5

**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:

*\$ wget https://downloads.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz*

```
hadoop@sudharsan-sundar-VirtualBox:~$ wget https://archive.apache.org/dist/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz
--2024-09-20 01:21:20-- https://archive.apache.org/dist/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 278813748 (266M) [application/x-gzip]
Saving to: 'apache-hive-3.1.2-bin.tar.gz'

apache-hive-3.1.2-bin.tar.gz  6%[==>] 18.50M  228KB/s  eta 13m 47s
```

*\$ tar -xvf apache-hive-3.1.2-bin.tar.gz*

```
hadoop@sudharsan-sundar-VirtualBox:~$ tar -xvf apache-hive-3.1.2-bin.tar.gz
apache-hive-3.1.2-bin/LICENSE
apache-hive-3.1.2-bin/NOTICE
apache-hive-3.1.2-bin/RELEASE_NOTES.txt
apache-hive-3.1.2-bin/binary-package-licenses/asm-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/com.google.protobuf-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/com.ibm.icu.icu4j-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/com.sun.jersey-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/com.thoughtworks.paranamer-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/javax.transaction.transaction-api-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/javolution-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/jline-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/NOTICE
apache-hive-3.1.2-bin/binary-package-licenses/org.abego.treelayout.core-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/org antlr-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/org antlr antlr4-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/org antlr stringtemplate-LICENSE
```

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

In this step, we are going to do two things

- o 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
```

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 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:
```

### 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 MySQL by 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';
```

```
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>
```

### 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](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
```

```
hadoop@sudharsan-sundar-VirtualBox:~$ hadoop fs -ls /user/hive/warehouse/
hadoop@sudharsan-sundar-VirtualBox:~$ hadoop fs -chmod g+w /tmp
hadoop@sudharsan-sundar-VirtualBox:~$ hadoop fs -chmod g+w /user/hive/warehouse
hadoop@sudharsan-sundar-VirtualBox:~$ schematool -initSchema -dbType derby
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/hive/lib/log4j-slf4j-impl-2.17.1.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:derby;;databaseName=metastore_db;create=true
Metastore Connection Driver : org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User: APP
Starting metastore schema initialization to 3.1.0
Initialization script hive-schema-3.1.0.derby.sql
```

**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@sudharsan-sundar-VirtualBox:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/hive/lib/log4j-slf4j-impl-2.17.1.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]
Hive Session ID = 25c9b9bb-a89a-4863-95f5-4c9925f2874c

Logging initialized using configuration in jar:file:/home/hadoop/hive/lib/hive-common-3.1.3.jar!/hive-log4j2.properties
Async: true
Hive Session ID = eae0e522-e10d-4ef4-a7ad-c9a80535bb62
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;
```

**Result:**

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