

Installation of HIVE with MySQL

Download and install hive, Open the url : <https://mirrors.estointernet.in/apache/hive/> and download or Open the terminal and download by following below command through the terminal

```
hduser@bigdata-VirtualBox:~$ wget
https://mirrors.estointernet.in/apache/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz

hduser@bigdata-VirtualBox:~$ wget https://mirrors.estointernet.in/apache/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz
--2020-12-07 20:43:59-- https://mirrors.estointernet.in/apache/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz
Resolving mirrors.estointernet.in (mirrors.estointernet.in)... 43.255.166.254, 2403:8940:3:1::f
Connecting to mirrors.estointernet.in (mirrors.estointernet.in)[43.255.166.254]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 278813748 (266M) [application/octet-stream]
Saving to: 'apache-hive-3.1.2-bin.tar.gz'

apache-hive-3.1.2-b 100%[=====] 265.90M  2.05MB/s   in 2m 2s
2020-12-07 20:46:01 (2.18 MB/s) - 'apache-hive-3.1.2-bin.tar.gz' saved [278813748/278813748]
```

Extract the hive tar file by type following below command

```
hduser@bigdata-VirtualBox:~$ tar -xf apache-hive-3.1.2-bin.tar.gz
```

```
hduser@bigdata-VirtualBox:~$ tar -xf apache-hive-3.1.2-bin.tar.gz
```

Update the hive configuration properties in bashrc file edit and update bashrc file using below command to edit.

```
hduser@bigdata-VirtualBox:~$ nano .bashrc
```

Add below properties as follows

```
export HIVE_HOME=/home/hduser/apache-hive-3.1.2-bin
export PATH=$PATH:$HIVE_HOME/bin
```

```
GNU nano 2.9.3 .bashrc
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

#Hadoop Related Options
export HADOOP_HOME=/home/hduser/hadoop-3.2.1
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"

#Hive related variables
export HIVE_HOME=/home/hduser/apache-hive-3.1.2-bin
export PATH=$PATH:$HIVE_HOME/bin
```

And to save the file after adding above properties

```
hduser@bigdata-VirtualBox:~$ source ~/.bashrc
```

Goto apache hive/bin folder and update the path where you install the hadoop in the hive-config.sh file

```
hduser@bigdata-VirtualBox:~/apache-hive-3.1.2-bin/bin$ nano hive-config.sh
hduser@bigdata-VirtualBox:~/apache-hive-3.1.2-bin/bin$
export HADOOP_HOME=/home/hduser/hadoop-3.2.1

GNU nano 2.9.3 hive-config.sh

check to see if the conf dir is given as an optional argument
while [ $# -gt 0 ]; do # Until you run out of parameters . . .
    case "$1" in
        --config)
            shift
            confdir=$1
            shift
            HIVE_CONF_DIR=$confdir
            ;;
        --auxpath)
            shift
            HIVE_AUX_JARS_PATH=$1
            shift
            ;;
        *)
            break;
            ;;
    esac
done

Allow alternate conf dir location.
HIVE_CONF_DIR="${HIVE_CONF_DIR:-$HIVE_HOME/conf}"

export HIVE_CONF_DIR=$HIVE_CONF_DIR
export HIVE_AUX_JARS_PATH=$HIVE_AUX_JARS_PATH

Default to use 256MB
export HADOOP_HEAPSIZE=${HADOOP_HEAPSIZE:-256}

export HADOOP_HOME=/home/hduser/hadoop-3.2.1
```

Check that hadoop is running or not if not then start the hadoop using below command

```
hduser@bigdata-VirtualBox:~/hadoop-3.2.1/sbin$ ./start-all.sh
```

To interact with hadoop we need to create directories in hadoop as follows below and give root permissions to them

```
hduser@bigdata-VirtualBox:~/hadoop-3.2.1/bin$ ./hadoop fs -mkdir /usr
hduser@bigdata-VirtualBox:~/hadoop-3.2.1/bin$ ./hadoop fs -mkdir /usr/hive/
hduser@bigdata-VirtualBox:~/hadoop-3.2.1/bin$ ./hadoop fs -mkdir
/usr/hive/warehouse
hduser@bigdata-VirtualBox:~/hadoop-3.2.1/bin$ ./hadoop fs -chmod g+w
/usr/hive/warehouse
```

Download and Install the mysql using below command

```
hduser@bigdata-VirtualBox:~$ sudo apt-get install mysql-server
```

Download mysql java connector and create a link between mysql and java using below commands

```
hduser@bigdata-VirtualBox:~$ sudo apt-get install libmysql-java
```

```
hduser@bigdata-VirtualBox:~$ ln -s /usr/share/java/mysql-connector-java.jar
/home/hduser/apache-hive-3.1.2-bin/lib/mysql-connector-java.jar
```

Configuring the mysql storage for hive

```
hduser@bigdata-VirtualBox:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights
reserved.
```

Creating username and password for MySQL, granting privileges using the following below commands

```
mysql> CREATE USER 'hduser'@'%' IDENTIFIED BY 'bigdata';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT all on *.* to 'hduser'@localhost identified by 'bigdata';
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

```
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

Configure the hive-site.xml

For that goto the hive / conf directory create hive-site.xml using below command

```
hduser@bigdata-VirtualBox:~/apache-hive-3.1.2-bin/conf$ sudo gedit
hive-site.xml
```

And add the below configurations in that file.

```
<configuration>
  <property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://localhost/metastore?createDatabaseIfNotExist=true&useSSL=false</value>
    <description>metadata is stored in a MySQL server</description>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.jdbc.Driver</value>
    <description>MySQL JDBC driver class</description>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>hduser</value>
    <description>user name for connecting to mysql server</description>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>bigdata</value>
    <description>password for connecting to mysql server</description>
  </property>
</configuration>
```

Remove the metastore db and initialize the schema in the mysql using below commands

```
hduser@bigdata-VirtualBox:~/apache-hive-3.1.2-bin$ rm -Rf
$HIVE_HOME/metastore_db
```

```
hduser@bigdata-VirtualBox:~/apache-hive-3.1.2-bin$ bin/schematool -dbType
```

```
mysql -initSchema
```

To run the hive goto hive/bin directory or hive then type the below command

```
hduser@bigdata-VirtualBox:~/apache-hive-3.1.2-bin$ ./bin/hive
```

```
hduser@bigdata-VirtualBox:~/apache-hive-3.1.2-bin$ ./bin/hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hduser/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/
SLF4J: Found binding in [jar:file:/home/hduser/hadoop-3.2.1/share/hadoop/common/lib/slf4j-log4j12-1.7.25.j
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 = 7f59fca4-079f-4392-bf8f-e6b24f502188

Logging initialized using configuration in jar:file:/home/hduser/apache-hive-3.1.2-bin/lib/hive-common-3.1
Hive Session ID = 4c6aaba5-3f95-43ee-92fd-a8f6081eb218
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a differ
g Hive 1.X releases.
hive> use de
decimal(      decode(      default      defined      degrees(      delimited      dense_rank(      desc
hive> use default
> ;
OK
Time taken: 2.952 seconds
hive> show tables;
OK
Time taken: 0.928 seconds
hive> █
```

Errors while installing hive :

```
hduser@bigdata-VirtualBox:~/hadoop-3.2.1/bin$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hduser/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/hduser/hadoop-3.2.1/share/hadoop/common/lib/slf4j-log4j12-1.7.25.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]
Exception in thread "main" java.lang.NoSuchMethodError: com.google.common.base.Preconditions.checkArgument(ZLjava/lang/String;Ljava/lang/Object;)V
    at org.apache.hadoop.conf.Configuration.set(Configuration.java:1357)
    at org.apache.hadoop.conf.Configuration.set(Configuration.java:1338)
    at org.apache.hadoop.mapred.JobConf.setJar(JobConf.java:536)
    at org.apache.hadoop.mapred.JobConf.setJarByClass(JobConf.java:554)
    at org.apache.hadoop.mapred.JobConf.<init>(JobConf.java:448)
    at org.apache.hadoop.hive.conf.HiveConf.initialize(HiveConf.java:5141)
    at org.apache.hadoop.hive.conf.HiveConf.<init>(HiveConf.java:5099)
    at org.apache.hadoop.hive.common.LogUtils.initHiveLog4jCommon(LogUtils.java:97)
    at org.apache.hadoop.hive.common.LogUtils.initHiveLog4j(LogUtils.java:81)
    at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:699)
    at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:683)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at org.apache.hadoop.util.RunJar.run(RunJar.java:323)
    at org.apache.hadoop.util.RunJar.main(RunJar.java:236)
```

Solution: To Fix the issue follow these steps

find out the version of the guava jar :

- Go to the \$HIVE_HOME/lib folder and find out the version of guava. For Hive 3.0.0, it is guava-19.0.jar.
- Go to \$HADOOP_HOME/share/hadoop/common/lib folder and out the version of guava. For Hadoop 3.2.1, the version is guava-27.0-jre.jar.
- If they are not the same (which is true for this case), delete the older version and copy the newer version in both.
- In this case, delete guava-19.0.jar in the Hive lib folder, and then copy guava-27.0-jre.jar from the Hadoop folder to Hive.

To remove jar file :

```
hduser@bigdata-VirtualBox:~/apache-hive-3.1.2-bin/lib$ rm guava-19.0.jar
```

Copy the latest version jar from hadoop lib directory to hive lib directory :

```
hduser@bigdata-VirtualBox:~/hadoop-3.2.1/share/hadoop/common/lib$ cp
guava-27.0-jre.jar /home/hduser/apache-hive-3.1.2-bin/lib/
```

