### **1.3 Install Apache Hive**

Download and extract Hive:

sh

wget https://dlcdn.apache.org/hive/hive-3.1.3/apache-hive-3.1.3-bin.tar.gz

tar -xvzf apache-hive-3.1.3-bin.tar.gz

sudo mv apache-hive-3.1.3-bin /opt/hive

Configure Hive environment:

sh

echo 'export HIVE\_HOME=/opt/hive' >> ~/.bashrc

echo 'export PATH=$HIVE\_HOME/bin:$PATH' >> ~/.bashrc

source ~/.bashrc

Check if Hive is installed:

sh

hive --version

### **1.4 Install MySQL**

Install MySQL server:

sh

sudo apt update

sudo apt install mysql-server -y

Start MySQL:

sh

sudo systemctl start mysql

sudo systemctl enable mysql

Check MySQL status:

sh

systemctl status mysql

## **Step 2: Configure MySQL for Hive Metastore**

### **2.1 Create Hive Metastore Database in MySQL**

Log in to MySQL:

sudo mysql -u root -p

Run the following SQL commands:

sql

CREATE DATABASE metastore;

CREATE USER 'hiveuser'@'localhost' IDENTIFIED BY 'hivepassword';

GRANT ALL PRIVILEGES ON metastore.\* TO 'hiveuser'@'localhost';

FLUSH PRIVILEGES;

EXIT;

### **2.2 Install MySQL JDBC Connector**

Download the MySQL JDBC connector:

wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-8.0.33.tar.gz

tar -xvzf mysql-connector-java-8.0.33.tar.gz

Move the JAR file to Hive and Spark libraries:

sudo mv mysql-connector-java-8.0.33/mysql-connector-java-8.0.33.jar /opt/hive/lib/

sudo mv mysql-connector-java-8.0.33/mysql-connector-java-8.0.33.jar /opt/spark/jars/

## **Step 3: Configure Hive Metastore to Use MySQL**

Modify the **Hive Metastore configuration file**:

nano /opt/hive/conf/hive-site.xml

Add or update the following properties inside <configuration>:

<configuration>

<property>

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

<value>jdbc:mysql://localhost:3306/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>hiveuser</value>

</property>

<property>

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

<value>hivepassword</value>

</property>

<property>

<name>hive.metastore.uris</name>

<value>thrift://localhost:9083</value>

</property>

</configuration>

Save and exit (CTRL+X, then Y, then Enter).

## **Step 4: Initialize and Start Hive Metastore**

Run the following command to initialize the Hive Metastore schema in MySQL:

schematool -dbType mysql -initSchema

Start the Hive Metastore service:

hive --service metastore &

## **Step 5: Configure Spark to Use Hive Metastore**

Modify Spark’s default configurations:

nano /opt/spark/conf/spark-defaults.conf

Add the following lines:

spark.sql.catalogImplementation=hive

spark.hadoop.hive.metastore.uris=thrift://localhost:9083

Save and exit.

Copy the Hive hive-site.xml file to Spark:

sh

cp /opt/hive/conf/hive-site.xml /opt/spark/conf/

## **Step 6: Start Spark and Verify Hive Integration**

Start **PySpark** with Hive support:

pyspark --conf spark.sql.catalogImplementation=hive

Run the following commands in PySpark:

from pyspark.sql import SparkSession

spark = SparkSession.builder \

.appName("SparkHiveIntegration") \

.config("spark.sql.catalogImplementation", "hive") \

.enableHiveSupport() \

.getOrCreate()

# Show Hive databases

spark.sql("SHOW DATABASES").show()

Or start **Spark SQL**:

sh

spark-sql --conf spark.sql.catalogImplementation=hive

Then run:

sql

SHOW DATABASES;

You should see Hive databases listed, confirming Spark is using MySQL-backed Hive Metastore.

## **Step 7: Verify MySQL Stores Hive Metadata**

Check the Hive Metastore database in MySQL:

sh

sudo mysql -u root -p

USE metastore;

SHOW TABLES;

You should see tables like TBLS, DATABASES, and PARTITIONS, confirming that **Hive metadata is stored in MySQL**.