Created By: Abid Aziz
Email: AzizAbid1@proton.me
Phone: +8801642818018

HIVE INSTALLATION:

1. (Login as hadoopuser with password)
   Download Hive:
   wget https://apache.osuosl.org/hive/hive-3.1.3/apache-hive-3.1.3-bin.tar.gz
   Extract the file and move:
   tar xzf apache-hive-3.1.3-bin.tar.gz
   sudo mv apache-hive-3.1.3-bin /usr/lib/hive

2. sudo nano /.bashrc
   Add the following lines:
   # Hive configuration
   export HIVE_HOME=/usr/lib/hive/apache-hive-3.1.3-bin
   export PATH=$PATH:$HIVE_HOME/bin
   Source the updated `.bashrc` file: source /.bashrc
   Ensure the user has the necessary permissions to access Hive and Hadoop directories:
   sudo chown -R hadoopuser:hadoopuser /usr/lib/hive/apache-hive-3.1.3-bin
   sudo chmod -R 755 /usr/lib/hive/apache-hive-3.1.3-bin

3. Edit `hive-site.xml` to configure the metastore:
   Add the following configuration:

```
<configuration>
  <property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:derby:;databaseName=metastore_db;create=true</value>
    <description>JDBC connect string for a JDBC metastore</description>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>org.apache.derby.jdbc.EmbeddedDriver</value>
    <description>Driver class name for a JDBC metastore</description>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>APP</value>
    <description>Username to use against metastore database</description>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>mine</value>
    <description>Password to use against metastore database</description>
```

```
  </property>

  <property>
    <name>datanucleus.schema.autoCreateTables</name>
    <value>true</value>
    <description>Auto create tables</description>
  </property>

  <property>
    <name>hive.metastore.schema.verification</name>
    <value>false</value>
    <description>Enforce metastore schema version consistency</description>
  </property>
</configuration>
```

4. Initialize the Hive metastore schema:
   $HIVE_HOME/bin/schematool -dbType derby -initSchema –verbose

5. Use `nohup` to run the Hive metastore service in the background:
   nohup $HIVE_HOME/bin/hive --service metastore &
   Check if the metastore is running:
   ps aux | grep metastore (You should see only one process named metastore is running)

6. Start the Hive CLI to interact with Hive:
   Hive
   In the Hive CLI, verify the databases:
   show databases; (It should show OK, Default and time taken)

7. Create a New Database (Optional, Just to check the hive is running or not perfectly)
   CREATE DATABASE IF NOT EXISTS new_database;
   USE new_database;
   Create A Sample Hive Table:

```
CREATE TABLE IF NOT EXISTS sample_table (
  id INT,
  name STRING,
  age INT,
  address STRING,
  salary FLOAT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```

Example of Loading Data into the Table

Assuming you have a CSV file (`sample_data.csv`) with the following content:

1,John Doe,30,123 Main St,60000.0

2,Jane Smith,25,456 Elm St,75000.0

3,Bob Johnson,35,789 Oak St,50000.0

You can load the data into the table as follows:

LOAD DATA LOCAL INPATH '/path/to/sample_data.csv' INTO TABLE sample_table; (change /path/to/)

Querying the Table

To query the data in the table:

SELECT * FROM sample_table;

To get the total salary from the `sample_table`, you can use the `SUM` function in an SQL query. Here is the query you would use:

SELECT SUM(salary) AS total_salary FROM sample_table;

To rank the salaries in the `sample_table`, you can use the `RANK` function along with a `SELECT` query. Here's an example:

```
SELECT
  id,
  name,
  age,
  address,
  salary,
  RANK() OVER (ORDER BY salary DESC) AS salary_rank
FROM sample_table;
```

Common Issues and Solutions

- **Check Logs**: If you encounter issues, check the `nohup.out` log file for errors: tail -f nohup.out
- **Database Cleanup**: If you face persistent issues with Derby, ensure no other instances are running, and consider deleting existing `metastore_db` directories: rm -rf /path/to/metastore_db (Change /path/to according to your path)
- **Metastore Already Running**

  If the metastore is already running or another instance is trying to use the same Derby database, you might see errors related to Derby lock files or duplicate metastore instances.

  Solution:

  Ensure no other instances are running:

  ```
  ps aux | grep metastore
  ```

  - Kill any existing metastore processes:

  ```
  kill -9 <PID>
  ```

- **Schema Initialization Errors**

  Errors like `FUNCTION 'NUCLEUS_ASCII' already exists` indicate that the schema is partially initialized.

  Solution:

  - Remove the existing metastore database and re-initialize:

    ```
    rm -rf /usr/lib/hive/apache-hive-3.1.3-bin/metastore_db
    $HIVE_HOME/bin/schematool -dbType derby -initSchema --verbose
    ```

SPARK INSTALLATION:

1. su – hadoopuser

2. Download Scala:
   wget https://downloads.lightbend.com/scala/2.12.6/scala-2.12.6.tgz
   Extract and move the file:
   tar xvf scala-2.12.6.tgz
   sudo mv scala-2.12.6 /usr/local/scala

3. sudo nano /.bashrc
   Add the following line at the end of the file:
   # Scala configuration
   export SCALA_HOME=/usr/local/scala
   export PATH=$PATH:$SCALA_HOME/bin
   Source the updated `.bashrc` file: source /.bashrc

4. Download Apache Spark:
   wget https://downloads.apache.org/spark/spark-3.3.2/spark-3.3.2-bin-hadoop3.tgz
   Extract and move the file:
   tar xvf spark-3.3.2-bin-hadoop3.tgz
   sudo mv spark-3.3.2-bin-hadoop3 /usr/local/spark

5. sudo nano /.bashrc
   Add the following line at the end of the file:
   # Spark configuration
   export SPARK_HOME=/usr/local/spark
   export PATH=$PATH:$SPARK_HOME/bin
   Source the file to save changes: source /.bashrc

6. Start the Spark shell to verify the installation:
   spark-shell
   You should see the Spark shell prompt (‘`scala>`’).
   Try performing 2+2 there to check if it’s working
   Exit the Spark shell: scala> :quit

7. To ensure PySpark is installed and working:
   Run PySpark: pyspark
   You should see the PySpark shell prompt (‘`>>>`’).
   Exit PySpark: >>> exit()