**Step-by-Step Procedure**

**Step 1: Install Java**

1. **Download JDK**:
   * Go to the Oracle JDK download page or adopt OpenJDK.
   * Download and install JDK 8 (e.g., jdk-8u351-windows-x64.exe).
2. **Set JAVA\_HOME**:
   * After installation, locate the JDK installation path (e.g., C:\Program Files\Java\jdk1.8.0\_351).
   * Set the environment variable:
     + Right-click 'This PC' > Properties > Advanced system settings > Environment Variables.
     + Under System Variables, click 'New' and add:
       - Variable name: JAVA\_HOME
       - Variable value: C:\Program Files\Java\jdk1.8.0\_351
     + Add %JAVA\_HOME%\bin to the Path variable.
3. **Verify Java**:
   * Open Command Prompt and run:

java -version

* + Ensure it shows the installed JDK version (e.g., java version "1.8.0\_351").

**Step 2: Install Hadoop**

1. **Download Hadoop**:
   * Download Hadoop (e.g., Hadoop 3.3.6) from the Apache Hadoop website.
   * Extract the tar.gz file to a directory (e.g., C:\hadoop-3.3.6).
2. **Set HADOOP\_HOME**:
   * Set the environment variable:
     + Variable name: HADOOP\_HOME
     + Variable value: C:\hadoop-3.3.6
     + Add %HADOOP\_HOME%\bin and %HADOOP\_HOME%\sbin to the Path variable.
3. **Install Winutils**:
   * Hadoop on Windows requires winutils.exe (not included in the Hadoop distribution).
   * Download the winutils.exe for your Hadoop version from a reliable source (e.g., GitHub repositories like steveloughran/winutils).
   * Place winutils.exe in %HADOOP\_HOME%\bin.
4. **Configure Hadoop**:
   * Edit Hadoop configuration files in %HADOOP\_HOME%\etc\hadoop:
     + **core-site.xml**:

<**configuration**>

<**property**>

<**name**>fs.defaultFS</**name**>

<**value**>hdfs://localhost:9000</**value**>

</**property**>

</**configuration**>

* + - **hdfs-site.xml**:

<**configuration**>

<**property**>

<**name**>dfs.replication</**name**>

<**value**>1</**value**>

</**property**>

<**property**>

<**name**>dfs.namenode.name.dir</**name**>

<**value**>file:///C:/hadoop-3.3.6/data/namenode</**value**>

</**property**>

<**property**>

<**name**>dfs.datanode.data.dir</**name**>

<**value**>file:///C:/hadoop-3.3.6/data/datanode</**value**>

</**property**>

</**configuration**>

* + - **mapred-site.xml**:

<**configuration**>

<**property**>

<**name**>mapreduce.framework.name</**name**>

<**value**>yarn</**value**>

</**property**>

</**configuration**>

* + - **yarn-site.xml**:

<**configuration**>

<**property**>

<**name**>yarn.nodemanager.aux-services</**name**>

<**value**>mapreduce\_shuffle</**value**>

</**property**>

</**configuration**>

1. **Create Data Directories**:
   * Create directories for NameNode and DataNode (e.g., C:\hadoop-3.3.6\data\namenode and C:\hadoop-3.3.6\data\datanode).
2. **Format NameNode**:
   * Open Command Prompt and run:

%HADOOP\_HOME%\bin\hdfs namenode -format

1. **Start Hadoop Services**:
   * Run:

%HADOOP\_HOME%\sbin\start-dfs.cmd

%HADOOP\_HOME%\sbin\start-yarn.cmd

* + Verify Hadoop is running by accessing:
    - NameNode: http://localhost:9870
    - YARN: http://localhost:8088

1. **Verify Hadoop**:
   * Run:

%HADOOP\_HOME%\bin\hadoop fs -ls /

* + Ensure no errors occur, and you see the HDFS root directory.

**Step 3: Install a Bash Environment**

1. **Install Git Bash or Cygwin**:
   * Download and install Git Bash (preferred for simplicity) or Cygwin.
   * This provides a Unix-like environment to run Hive scripts, as Hive’s shell scripts are not natively compatible with Windows Command Prompt.
2. **Verify Bash**:
   * Open Git Bash and run:

bash --version

**Step 4: Install Apache Hive**

1. **Download Hive**:
   * Download Apache Hive (e.g., Hive 3.1.3) from the Apache Hive website.
   * Extract the tar.gz file to a directory (e.g., C:\hive-3.1.3).
2. **Set HIVE\_HOME**:
   * Set the environment variable:
     + Variable name: HIVE\_HOME
     + Variable value: C:\hive-3.1.3
     + Add %HIVE\_HOME%\bin to the Path variable.
3. **Configure Hive**:
   * Navigate to %HIVE\_HOME%\conf.
   * Copy hive-default.xml.template to hive-site.xml.
   * Edit hive-site.xml to configure the metastore and Hadoop paths:

<**configuration**>

<**property**>

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

<**value**>jdbc:derby:;databaseName=C:/hive-3.1.3/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**>hive.metastore.warehouse.dir</**name**>

<**value**>/user/hive/warehouse</**value**>

<**description**>Location of default database for the warehouse</**description**>

</**property**>

<**property**>

<**name**>hive.metastore.schema.verification</**name**>

<**value**>false</**value**>

</**property**>

</**configuration**>

* + Note: This uses Derby as the metastore for simplicity. For production, consider MySQL or PostgreSQL (requires additional setup).

1. **Set Hadoop Path in Hive**:
   * Ensure Hive can access Hadoop by setting:

export HADOOP\_HOME=C:/hadoop-3.3.6

in Git Bash or add it to %HIVE\_HOME%\bin\hive-config.sh (replace backslashes with forward slashes).

**Step 5: Initialize Hive Metastore**

1. **Initialize Derby Metastore**:
   * Open Git Bash and navigate to %HIVE\_HOME%\bin.
   * Run:

./schematool -initSchema -dbType derby

* + This creates the metastore database in C:\hive-3.1.3\metastore\_db.

**Step 6: Start Hive**

1. **Start Hadoop Services**:
   * Ensure Hadoop services are running (see Step 2.7).
2. **Create Hive Warehouse Directory in HDFS**:
   * Run in Command Prompt or Git Bash:

%HADOOP\_HOME%\bin\hadoop fs -mkdir -p /user/hive/warehouse

%HADOOP\_HOME%\bin\hadoop fs -chmod -R 777 /user/hive/warehouse

1. **Launch Hive**:
   * In Git Bash, navigate to %HIVE\_HOME%\bin and run:

./hive

* + You should see the Hive CLI prompt (hive>).

1. **Test Hive**:
   * Run a simple query to verify:

CREATE TABLE test\_table (id INT, name STRING);

SHOW TABLES;

* + If test\_table appears, Hive is set up correctly.

**Step 7: Optional - Configure MySQL Metastore (Production)**

1. **Install MySQL**:
   * Download and install MySQL Community Server.
   * Set up a MySQL user and database (e.g., hive database, user hiveuser, password hivepass).
2. **Download MySQL JDBC Driver**:
   * Download the MySQL Connector/J from the MySQL website.
   * Place the .jar file in %HIVE\_HOME%\lib.
3. **Update hive-site.xml**:
   * Replace the Derby configuration with:

<**property**>

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

<**value**>jdbc:mysql://localhost:3306/hive?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**>hivepass</**value**>

</**property**>

1. **Initialize MySQL Metastore**:
   * Run:

./schematool -initSchema -dbType mysql

**Step 8: Troubleshooting**

* **Hadoop Errors**: Ensure winutils.exe is in %HADOOP\_HOME%\bin and Hadoop services are running.
* **Hive Errors**:
  + Check %HIVE\_HOME%\logs for errors.
  + Verify JAVA\_HOME, HADOOP\_HOME, and HIVE\_HOME are correctly set.
  + Ensure the metastore database is accessible (Derby or MySQL).
* **Permission Issues**: Run Command Prompt or Git Bash as Administrator.
* **Path Issues**: Replace backslashes (\) with forward slashes (/) in configuration files for compatibility with Hive scripts.

**Step 9: Verify Installation**

1. Run Hive queries:

hive> CREATE TABLE sample (id INT, name STRING);

hive> INSERT INTO sample VALUES (1, 'Test');

hive> SELECT \* FROM sample;

1. Check HDFS for Hive data:

%HADOOP\_HOME%\bin\hadoop fs -ls /user/hive/warehouse/sample