Apache Hive 3.1.3

Installation & Configuration Guide

Hadoop HDFS + MapReduce + MySQL Metastore

VMware	17.0.1
Ubuntu Kylin	16.0.4.7
Hadoop	3.3.5
Spark	3.4.0
Hive	3.1.3
Kafka	2.12-3.5.1
MySQL Connector	5.1.40
Python	3.10.13

Author: Lee Yangweon (李阳源)
Date: 2026-03-08

1. Prerequisites (사전 준비사항)

Before installing Hive, ensure the following services are properly installed and running:

1.1 Required Services Verification

Hadoop HDFS and YARN must be running. Verify with these commands:

```
$ jps
  (NameNode, DataNode, SecondaryNameNode, ResourceManager, NodeManager 확인)

$ hdfs dfs -ls /
  (HDFS 정상 동작 확인)

$ hdfs dfsadmin -report
  (HDFS 클러스터 상태 확인)
```

1.2 MySQL Service Verification

MySQL must be running for Hive Metastore. Verify:

```
$ sudo service mysql status
$ mysql -u root -p -e "SELECT VERSION();"
```

2. Hive Download & Installation (Hive 다운로드 및 설치)

2.1 Download Apache Hive 3.1.3

Download the Hive 3.1.3 binary package:

```
$ cd /home/hadoop
$ wget https://archive.apache.org/dist/hive/hive-3.1.3/apache-hive-3.1.3-bin.tar.gz
```

2.2 Extract and Install

Extract to /usr/local and create a symbolic link:

```
$ sudo tar -xzf apache-hive-3.1.3-bin.tar.gz -C /usr/local/
$ sudo mv /usr/local/apache-hive-3.1.3-bin /usr/local/hive
$ sudo chown -R hadoop:hadoop /usr/local/hive
```

2.3 Set Environment Variables

Add the following to ~/.bashrc:

```
$ nano ~/.bashrc


# ===== Hive Environment =====
export HIVE_HOME=/usr/local/hive
export
PATH=$HIVE_HOME/bin:$JAVA_HOME/bin:$HADOOP_HOME/bin:$SPARK_HOME/bin:$KAFKA_HOME/bin
:$ANACONDA_HOME/bin:$PATH
export HIVE_CONF_DIR=$HIVE_HOME/conf


# Hive Beeline alias (MySQL metastore)
alias hive-bee='$HIVE_HOME/bin/beeline -u "jdbc:hive2://localhost:10000" -n
hiveuser -p 123456 --hiveconf hive.server2.logging.operation.level=NONE'
```

Apply the changes:

```
$ source ~/.bashrc
```

3. Guava JAR Conflict Resolution (Guava JAR 충돌 해결)

Hive 3.1.3 and Hadoop 3.3.5 use different versions of Guava library. This must be resolved:

```
# Check Hadoop Guava version
$ ls $HADOOP_HOME/share/hadoop/common/lib/guava-*.jar
  (e.g., guava-27.0-jre.jar)
```

내 시스템결과 : /usr/local/hadoop/share/hadoop/common/lib/guava-27.0-jre.jar

```
# Check Hive Guava version
$ ls $HIVE_HOME/lib/guava-*.jar
  (e.g., guava-19.0.jar)
```

내 시스템결과 : /usr/local/hive/lib/guava-19.0.jar

```
# Replace Hive Guava with Hadoop Guava
$ rm $HIVE_HOME/lib/guava-19.0.jar
$ cp $HADOOP_HOME/share/hadoop/common/lib/guava-27.0-jre.jar $HIVE_HOME/lib/
```

Note: Without this step, Hive will fail with java.lang.NoSuchMethodError related to Guava.

4. MySQL Configuration for Hive Metastore (MySQL 설정)

4.1 Create Hive Metastore Database and User

Log into MySQL as root and create the database and user:

```
$ mysql -u root -p

mysql> CREATE DATABASE IF NOT EXISTS hive_metastore
       DEFAULT CHARACTER SET utf8
       DEFAULT COLLATE utf8_general_ci;

mysql> CREATE USER 'hiveuser'@'localhost' IDENTIFIED BY '123456';

mysql> GRANT ALL PRIVILEGES ON hive_metastore.* TO 'hiveuser'@'localhost';

mysql> GRANT ALL PRIVILEGES ON hive_metastore.* TO 'hiveuser'@'%'
       IDENTIFIED BY '123456';

mysql> FLUSH PRIVILEGES;

mysql> EXIT;
```

4.2 Verify MySQL User

Test the hiveuser connection:

```
$ mysql -u hiveuser -p123456 -e "SHOW DATABASES;"
  (hive_metastore 데이터베이스가 보여야 함)
```

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| hive_metastore     |
+--------------------+
```

4.3 Copy MySQL JDBC Connector

Copy the MySQL connector JAR to Hive lib directory:

```
$ cp /usr/local/spark/jars/mysql-connector-java-5.1.40.jar $HIVE_HOME/lib/


# Verify
$ ls $HIVE_HOME/lib/mysql-connector-java-*.jar
```

없으면 추출 (현재 내 컴퓨터에는 tar 파일만 있음. 주의!!!)

cd /tmp

```
tar -xf /usr/local/spark/jars/mysql-connector-java-5.1.40.tar
cp ./mysql-connector-java-5.1.40/mysql-connector-java-5.1.40-bin.jar
/usr/local/spark/jars/mysql-connector-java-5.1.40.jar
```

5. Hive Configuration Files (하이브 설정 파일)

5.1 Create hive-site.xml

Create the main Hive configuration file:

```
$ nano $HIVE_HOME/conf/hive-site.xml
```

Add the following content:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>

  <!-- MySQL Metastore Connection -->
  <property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://localhost:3306/hive_metastore?
createDatabaseIfNotExist=true&amp;useSSL=false</value>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.jdbc.Driver</value>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>hiveuser</value>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>123456</value>
  </property>

  <!-- Hive Metastore Configuration -->
  <property>
    <name>hive.metastore.warehouse.dir</name>
    <value>/user/hive/warehouse</value>
  </property>

  <property>
    <name>hive.metastore.schema.verification</name>
    <value>false</value>
  </property>

  <property>
    <name>datanucleus.schema.autoCreateAll</name>
```

```
  <value>true</value>
</property>

<!-- HiveServer2 Configuration -->
<property>
  <name>hive.server2.thrift.port</name>
  <value>10000</value>
</property>

<property>
  <name>hive.server2.thrift.bind.host</name>
  <value>localhost</value>
</property>

<property>
  <name>hive.server2.enable.doAs</name>
  <value>false</value>
</property>

<!-- Hive Execution Engine -->
<property>
  <name>hive.execution.engine</name>
  <value>mr</value>
</property>

<!-- Local mode auto -->
<property>
  <name>hive.exec.mode.local.auto</name>
  <value>true</value>
</property>

<!-- MapReduce Memory Settings -->
 <property>
  <name>mapreduce.map.memory.mb</name>
  <value>1024</value>
 </property>

 <property>
  <name>mapreduce.reduce.memory.mb</name>
  <value>1024</value>
 </property>

 <property>
  <name>mapreduce.map.java.opts</name>
  <value>-Xmx768m</value>
```

```
    </property>

    <property>
      <name>mapreduce.reduce.java.opts</name>
      <value>-Xmx768m</value>
    </property>

    <!-- Local mode for small jobs (avoid YARN overhead) -->
    <property>
      <name>hive.exec.mode.local.auto</name>
      <value>true</value>
    </property>

    <property>
      <name>hive.exec.mode.local.auto.inputbytes.max</name>
      <value>134217728</value>
    </property>

</configuration>
```

- 반드시 필요

```
메시지 레벨 조정 (info →error)
cd $HIVE_HOME/conf

# 템플릿이 있으면 복사
cp hive-log4j2.properties.template hive-log4j2.properties
# 편집
vim hive-log4j2.properties
# 수정 전
property.hive.log.level = INFO
# 수정 후
property.hive.log.level = ERROR
cp hive-env.sh.template hive-env.sh
다음 내용 추가:
export HADOOP_HEAPSIZE=1024
export HIVE_SERVER2_HEAPSIZE=1024
export HADOOP_CLIENT_OPTS="-Xmx1024m $HADOOP_CLIENT_OPTS"
```

5.2 Initialize Hive Metastore Schema

Initialize the Hive metastore schema in MySQL:

```
$ cd $HIVE_HOME
$ schematool -dbType mysql -initSchema
```

Expected output: 'schemaTool completed' message. If you see errors about existing tables, the schema may have been initialized before.

To re-initialize (if needed):

```
$ schematool -dbType mysql -initSchema -verbose


만일 여기서 fail이 나면 다음 내용을 수정한다.

hive-site.xml에서 ConnectionURL의 <value> 값을 반드시 한 줄로 수정하세요:

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

<value>jdbc:mysql://localhost:3306/hive_metastore?createDatabaseIfNotExist=true&amp
;useSSL=false</value>
</property>

수정한 뒤, 이전 시도에서 생성된 불완전한 테이블들을 먼저 정리하고 다시 초기화합니다:

# 1. MySQL에서 기존 불완전한 스키마 삭제

mysql -u hiveuser -p123456 -e "DROP DATABASE IF EXISTS hive_metastore; CREATE
DATABASE hive_metastore DEFAULT CHARACTER SET utf8 DEFAULT COLLATE
utf8_general_ci;"


# 2. 다시 초기화

schematool -dbType mysql -initSchema


schemaTool completed 메시지가 나오면 성공입니다.
```

6. HDFS Directory Setup (HDFS 디렉토리 설정)

Create required HDFS directories for Hive:

```
$ hdfs dfs -mkdir -p /user/hive/warehouse
$ hdfs dfs -mkdir -p /tmp
$ hdfs dfs -chmod g+w /user/hive/warehouse
$ hdfs dfs -chmod g+w /tmp


# Verify
$ hdfs dfs -ls /user/hive/
$ hdfs dfs -ls /tmp
```

7. SLF4J Conflict Resolution (SLF4J 충돌 해결)

Hive includes a log4j-slf4j binding that conflicts with Hadoop. Remove it:

```
$ cd $HIVE_HOME/lib
$ mv log4j-slf4j-impl-2.17.1.jar log4j-slf4j-impl-2.17.1.jar.bak
```

This prevents the 'SLF4J: Class path contains multiple SLF4J bindings' warning.

8. Start and Test Hive (Hive 구동 및 테스트)

8.1 Start Hive Metastore Service

Start the Hive Metastore service in background:

```
$ nohup hive --service metastore > /tmp/hive-metastore.log 2>&1 &


# Wait about 10 seconds, then verify port 9083
$ netstat -tln | grep 9083
  (Port 9083 should be LISTENING)
```

```
(base) hadoop@hadoop01:/usr/local/hive/lib$ netstat -tln |grep 9083
tcp6       0      0 :::9083                 :::*                    LISTEN
```

8.2 Start HiveServer2

Start HiveServer2 for Beeline connections:

```
$ nohup hive --service hiveserver2 > /tmp/hive-server2.log 2>&1 &


# Wait about 15-20 seconds, then verify port 10000 (sleep 20)
$ netstat -tln | grep 10000
  (Port 10000 should be LISTENING)
```

```
(base) hadoop@hadoop01:/usr/local/hive/lib$ netstat -tln |grep 10000
tcp6       0      0 :::10000                :::*                    LISTEN
```

8.3 Connect with Beeline (using alias)

Use the configured alias to connect:

```
$ hive-bee
(beeline이 spark도 있음. 따라서 hive path가 앞에 나와야 됨.)
# This is equivalent to:
$ beeline -u "jdbc:hive2://localhost:10000" -n hiveuser -p 123456
```

```
(base) hadoop@hadoop01:~$ netstat -tln |grep -E "9083|10000"
tcp6       0      0 :::9083                 :::*                    LISTEN
tcp6       0      0 :::10000                :::*                    LISTEN
(base) hadoop@hadoop01:~$ hive-bee
Connecting to jdbc:hive2://localhost:10000
Connected to: Apache Hive (version 3.1.3)
Driver: Hive JDBC (version 3.1.3)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.3 by Apache Hive
0: jdbc:hive2://localhost:10000>
```

8.4 Test Hive Queries in Beeline

Run test queries to verify everything works:

```
0: jdbc:hive2://localhost:10000> SHOW DATABASES;

0: jdbc:hive2://localhost:10000> CREATE DATABASE test_db;

0: jdbc:hive2://localhost:10000> USE test_db;

0: jdbc:hive2://localhost:10000> CREATE TABLE test_table (
    id INT,
    name STRING,
    age INT
  ) ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ','
  STORED AS TEXTFILE;
```

[터미널]

```
echo -e "1,alice,25\n2,lee,30\n3,kim,35" > /tmp/test_data.csv
LOAD DATA LOCAL INPATH '/tmp/test_data.csv' INTO TABLE test_table; SELECT * FROM
test_table;

0: jdbc:hive2://localhost:10000> INSERT INTO test_table VALUES
  (1, 'Alice', 25), (2, 'Bob', 30), (3, 'Charlie', 35);

0: jdbc:hive2://localhost:10000> SELECT * FROM test_table;

0: jdbc:hive2://localhost:10000> !quit
```

9. Installation Verification Checklist (설치 확인 체크리스트)

Run these commands to verify the complete installation:

```
# 1. Check Hive version
$ hive --version
```

```
(base) hadoop@hadoop01:~$ hive --version
Hive 3.1.3
Git git://MacBook-Pro.fios-router.home/Users/ngangam/commit/hive -r 4df4d75bf1e16fe0af75aad0b4179c34c34c07fc975
Compiled by ngangam on Sun Apr 3 16:58:16 EDT 2022
From source with checksum 5da234766db5dfbe3e92926c9bbab2af
```

```
# 2. Check environment variables
$ echo $HIVE_HOME
$ echo $PATH | tr ':' '\n' | grep hive
```

```
(base) hadoop@hadoop01:~$ echo $HIVE_HOME
/usr/local/hive
(base) hadoop@hadoop01:~$ echo $PATH | tr ':' '\n' | grep hive
/usr/local/hive/bin
/usr/local/hive/bin
/usr/local/hive/bin
/usr/local/hive/bin
/usr/local/hive/bin
/usr/local/hive/bin
/usr/local/hive/bin
/usr/local/hive/bin
/usr/local/hive/bin
/usr/local/hive/bin
/usr/local/hive/bin
/usr/local/hive/bin
/usr/local/hive/bin
/usr/local/hive/bin
/usr/local/hive/bin
```

```
# 3. Check MySQL metastore database
$ mysql -u hiveuser -p123456 -e "USE hive_metastore; SHOW TABLES;"
```

```
(base) hadoop@hadoop01:~$ mysql -u hiveuser -p123456 -e "USE hive_metastore; SHOW TABLES;"
mysql: [Warning] Using a password on the command line interface can be insecure.
+-----------------------------+
| Tables_in_hive_metastore    |
+-----------------------------+
| AUX_TABLE                   |
| BUCKETING_COLS              |
| CDS                         |
| COLUMNS_V2                  |
| COMPACTION_QUEUE            |
```

```
# 4. Check HDFS directories
$ hdfs dfs -ls /user/hive/warehouse
```

```
(base) hadoop@hadoop01:~$ hdfs dfs -ls /user/hive/warehouse
Found 1 items
drwxr-xr-x   - hadoop supergroup          0 2026-03-08 21:18 /user/hive/warehouse/test_db.db
(base) hadoop@hadoop01:~$
```

```
# 5. Check Hive Metastore port (9083)
$ netstat -tln | grep 9083
```

```
(base) hadoop@hadoop01:~$ netstat -tln | grep 9083
tcp6       0      0 :::9083                 :::*                    LISTEN
(base) hadoop@hadoop01:~$ netstat -tln | grep 10000
tcp6       0      0 :::10000                :::*                    LISTEN
```

```
# 6. Check HiveServer2 port (10000)
$ netstat -tln | grep 10000
```

```
# 7. Check JPS for RunJar processes (Hive services)
$ jps | grep RunJar
```

```
(base) hadoop@hadoop01:~$ jps |grep RunJar
15625 RunJar
16437 RunJar
16279 RunJar
10879 RunJar
(base) hadoop@hadoop01:~$
```

```
# 8. Test Beeline connection with alias
$ hive-bee
  (Should connect to jdbc:hive2://localhost:10000)
```

```
(base) hadoop@hadoop01:~$ hive-bee
Connecting to jdbc:hive2://localhost:10000
Connected to: Apache Hive (version 3.1.3)
Driver: Hive JDBC (version 3.1.3)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.3 by Apache Hive
0: jdbc:hive2://localhost:10000>
```

```
# 9. Check all services summary
$ jps
  (NameNode, DataNode, SecondaryNameNode,
   ResourceManager, NodeManager, RunJar x2)
```

```
(base) hadoop@hadoop01:~$ jps
4096 SecondaryNameNode
5539 Kafka
15625 RunJar
4521 ResourceManager
3436 NameNode
4653 NodeManager
17453 Jps
3571 DataNode
16437 RunJar
16279 RunJar
6103 HistoryServer
6456 Main
4925 QuorumPeerMain
10879 RunJar
(base) hadoop@hadoop01:~$
```

10. Troubleshooting (문제 해결)

10.1 Common Errors

Error	Cause	Solution
Guava NoSuchMethodError	JAR version mismatch	Replace Hive guava with Hadoop version
MetaException	Schema not initialized	Run schematool -initSchema
Connection refused 10000	HiveServer2 not started	Start HiveServer2 and wait 20s
SLF4J binding conflict	Duplicate SLF4J JARs	Remove log4j-slf4j-impl from Hive lib
Permission denied HDFS	Wrong HDFS permissions	chmod g+w on /user/hive/warehouse

10.2 Log File Locations

```
Hive Metastore:  /tmp/hive-metastore.log
HiveServer2:     /tmp/hive-server2.log
Hive general:    /tmp/hadoop/hive.log
```

百见不如一打! 我们一起努力吧！

Professor: Lee Yangweon (李阳源)