

# HIVE 설치

1. 하이브 다운로드
2. OS 사용자 환경 설정
3. Metadata Store(MySQL) 설정
  - 3.1 MySQL 설치
  - 3.2 MySQL DB 및 사용자 생성
  - 3.3 MySQL JDBC Connector/J 설치
4. hive-site.xml 설정
5. HDFS 디렉토리 생성
6. Metastore 초기화

- 참고 URL
  - <https://sparkdia.tistory.com/12>
  - <https://jyoonddev.tistory.com/69>

## 1. 하이브 다운로드

- 하이브 다운로드 링크 : <https://downloads.apache.org/hive/>
- 2.3.9버전 다운로드

```
root@DESKTOP-SJTS912:/home/hyerimlee$ wget https://downloads.apache.org/hive/hive-2.3.9/apache-hive-2.3.9-bin.tar.gz
```

- 압축 풀기

```
root@DESKTOP-SJTS912:/home/hyerimlee$ tar -zxf apache-hive-2.3.9-bin.tar.gz
```

- 디렉토리 변경
  - /usr/local 로 디렉토리 변경

```
su hduser
sudo mv 'apache-hive-2.3.9-bin' /usr/local
```

- Hive의 소유자 변경

```
sudo chown -R hduser:hadoop /usr/local/apache-hive-2.3.9-bin/
```

- 파일 변경 반영

```
source ~/.bashrc
```

- [hive-env.sh](#) 파일 생성
  - 처음 hive를 설치하면 conf 디렉터리에는 템플릿 파일만 있기 때문

```
cd $HIVE_HOME
mv conf/hive-env.sh.template conf/hive-env.sh
```

- hive-env.sh에 하둡 홈 디렉터리를 설정
  - 하이브는 하둡 환경설정 정보를 이용해 하이브 질의를 수행하기 때문

```
# HADOOP_HOME=${bin}/../..../hadoop
HADOOP_HOME=/usr/local/hadoop
```

## 2. OS 사용자 환경 설정

```
sudo vi ~/.bashrc
```

```
export Hive_Home=/usr/local/apache-hive-2.3.9-bin
```

```
PATH="$HOME/.local/bin:$HOME/bin:$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$HIVE_HOME/bin"
```

```
export PATH
```

```
export HIVE_HOME=/usr/local/apache-hive-2.3.9-bin
PATH="$HOME/.local/bin:$HOME/bin:$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$HIVE_HOME/bin"
export PATH
```

- 파일 변경 반영

```
source ~/.bashrc
```

## 3. Metadata Store(MySQL) 설정

- 하이브는 기본적으로 아파치 더비데이브를 메타스토어로 사용하며, 다른 DBMS를 메타스토어로 사용하려면 DBMS와 JDBC Connector를 먼저 설치해야 함.
- 설치가 완료되면 메타스토어로 사용할 데이터베이스와 사용자 계정을 생성하고 hive-site.xml을 수정해야 함.
- 또한, DBMS의 JDBC 드라이버를 하이브의 lib 디렉터리에 복사해야 함.
  - 하이브는 JDBC를 이용해 각 DBMS에 접속하기 때문

### 3.1 MySQL 설치

- 하이브와의 안정적인 호환성을 위해 5.7.33 버전 설치
- 설치 URL : <https://downloads.mysql.com/archives/community/>

```
wget https://downloads.mysql.com/archives/get/p/23/file/mysql-server_5.7.33-1ubuntu18.04_amd64.deb-bundle.tar
```

- 압축해제 및 설치

```
tar -xvf mysql-server_5.7.33-1ubuntu18.04_amd64.deb-bundle.tar
```

- MySQL 패키지 설치

```
sudo apt install mysql-server-5.7
sudo mysql-server-core-5.7 mysql-client-5.7
sudo mysql-client-core-5.7
```

### 3.2 MySQL DB 및 사용자 생성

- root 계정으로 로그인

```
mysql -uroot -p
```

- 사용자 계정 doop 생성.

- 비밀번호를 입력한 뒤 mysql> 콘솔이 나오면, 다음 커맨드를 입력.

```
mysql> create user 'hive'@'%' identified by 'gdisme22!';
```

- 해당 계정에 루트 권한 부여

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'hive'@'%' IDENTIFIED BY 'gdisme22!';
mysql> flush privileges;
```

- Metastore로 사용할 데이터베이스명을 'metastore'로 정의, Hive의 기본 캐릭터셋이 utf8이기 때문에 생성할 데이터베이스의 기본 캐릭터셋도 utf-8로 설정

```
CREATE DATABASE metastore DEFAULT CHARACTER SET utf8;
```

- MySQL 사용자 계정 조회를 통해 정상적으로 완료되었는지 확인
  - 사용자 계정 조회 확인

```
SELECT HOST,USER FROM mysql.user;
```

```
mysql> SELECT HOST,USER FROM mysql.user;
+-----+-----+
| HOST      | USER                |
+-----+-----+
| doop01    | doop                 |
| localhost | debian-sys-maint     |
| localhost | mysql.session        |
| localhost | mysql.sys            |
| localhost | root                 |
+-----+-----+
5 rows in set (0.00 sec)
```

- 데이터베이스 목록 확인

```
mysql> SELECT SCHEMA_NAME,DEFAULT_CHARACTER_SET_NAME
-> FROM INFORMATION_SCHEMA.SCHEMATA;
```

```
mysql> SELECT SCHEMA_NAME,DEFAULT_CHARACTER_SET_NAME
-> FROM INFORMATION_SCHEMA.SCHEMATA;
+-----+-----+
| SCHEMA_NAME | DEFAULT_CHARACTER_SET_NAME |
+-----+-----+
| information_schema | utf8 |
| metastore         | utf8 |
| mysql             | latin1 |
| performance_schema | utf8 |
| sys               | utf8 |
+-----+-----+
5 rows in set (0.00 sec)
```

### 3.3 MySQL JDBC Connector/J 설치

- 설치 URL : <https://downloads.mysql.com/archives/c-j/>
- 8.0.24버전 설치

```
https://downloads.mysql.com/archives/get/p/3/file/mysql-connector-java-8.0.24.tar.gz
```

- 압축풀기

```
tar -zxf mysql-connector-java-8.0.24.tar.gz
```

- hive의 lib 디렉토리로 jar 파일을 옮겨줌

```
cd mysql-connector-java-8.0.24/  
mv mysql-connector-java-8.0.24.jar /usr/local/apache-hive-2.3.9-bin/lib
```

## 4. hive-site.xml 설정

- sudo vi \$HIVE\_HOME/conf/hive-site.xml

```
<?xml version="1.0"?>  
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  
<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.jdbc.Driver</value>  
  </property>  
  <property>  
    <name>javax.jdo.option.ConnectionUserName</name>  
    <value>hive</value>  
  </property>  
  <property>  
    <name>javax.jdo.option.ConnectionPassword</name>  
    <value>비밀번호</value>  
  </property>  
</configuration>
```

## 5. HDFS 디렉토리 생성

- 두 개의 디렉토리를 미리 생성하고 실행 권한도 설정해야 함
  - 하이버에서 업로드하는 데이터는 HDFS의 /user/hive/warehouse 에 저장됨.
  - 하이버에서 실행하는 잡의 여유공간으로 HDFSdml "/tmp/hive-유저명" 디렉토리를 사용.

```
cd /usr/local/hadoop/etc/hadoop  
start-all.sh  
hadoop dfs -mkdir /tmp  
hadoop dfs -mkdir -p /user/hive/warehouse  
hadoop dfs -chmod g+w /tmp  
hadoop dfs -chmod g+w /user/hive/warehouse  
hadoop dfs -chmod 777 /tmp/hive
```

## 6. Metastore 초기화

- 하이버를 실행하기 전에 하이버 메타스토어를 초기화해야함
  - dbType 뒤에 metastore로 설정한 DB를 적어주면 됨
  - 특별한 DB를 설정하지 않았다면, derby
  - mysql을 설정했다면 mysql

```
./bin/schematool -initSchema -dbType mysql
```

```
Starting metastore schema initialization to 2.3.0  
Initialization script hive-schema-2.3.0.mysql.sql  
Initialization script completed  
schemaTool completed
```

- 메타데이터 초기화 작업이 정상적으로 완료되면, hive에서 show databases를 실행하면 기본 데이터베이스인 default가 출력됨

```
./bin/hive  
hive> show databases;
```

```
hive> show databases;  
Loading class `com.mysql.j  
cally registered via the S  
OK  
default
```