

1. Hive 설치 및 환경설정

목 차

1. Hive 소개
2. Hive 설치
3. Hive 환경설정
4. Hadoop DFS 작업
5. Hive 실행 & 테이블 만들기

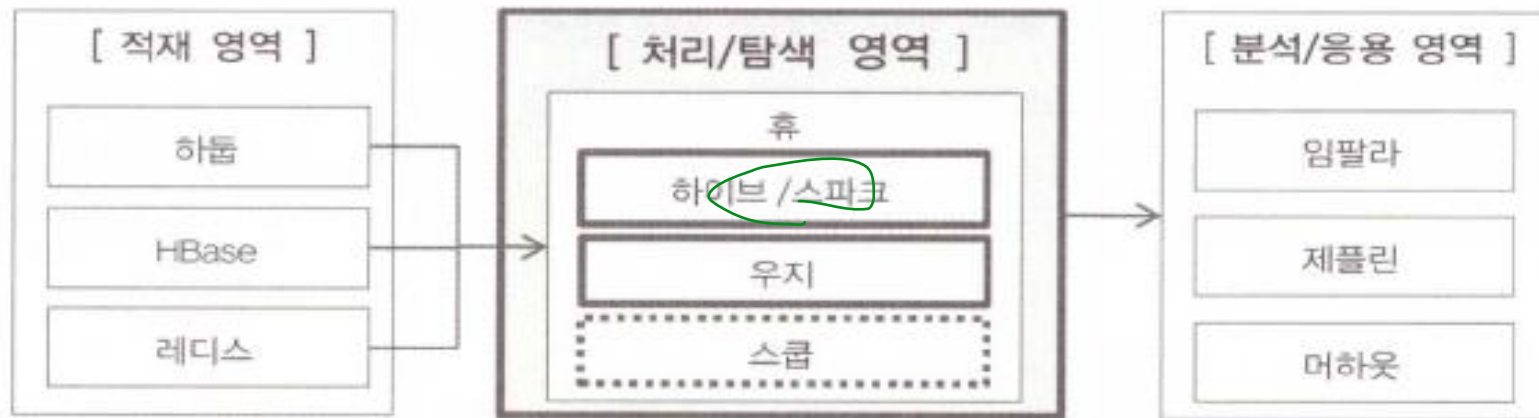
1. Hive 소개

- MapReduce는 복잡도가 높은 프로그래밍 기법이 요구되고, 이는 업무 분석가 및 관리자들에게 빅데이터 접근을 어렵게 만듦
- 이를 해결하기 위해 페이스북에서 SQL과 매우 유사한 방식으로 하둡 데이터에 접근성을 높인 Hive 개발
- 오픈 소스로 공개되면서 2016년 2월 하이버 2.0이 릴리스
- 빅데이터의 가장 대표적인 SQL on Hadoop 제품으로 자리 잡음

➤ Hive 주요 특징

- HiveQL이라고 하는 SQL과 유사한 쿼리를 사용
- HDFS 적재 데이터 이용 DW(data warehouse) 구축
- HiveQL로 정의한 내용을 Hive가 MapReduce 변환/ 실행
- Hive 구성요소에서 Metastore 존재
 - 이 부분에 DB 스키마 저장
 - 실제 데이터는 하둡의 HDFS에 분산되어 저장

처리/탐색 Layer



➤ 하이브/스파크

- 하둡에 적재된 데이터를 정제/변형/통합/분리/탐색 등의 작업 수행
- 데이터를 정형화된 구조로 정규화해 데이터 마트 생성

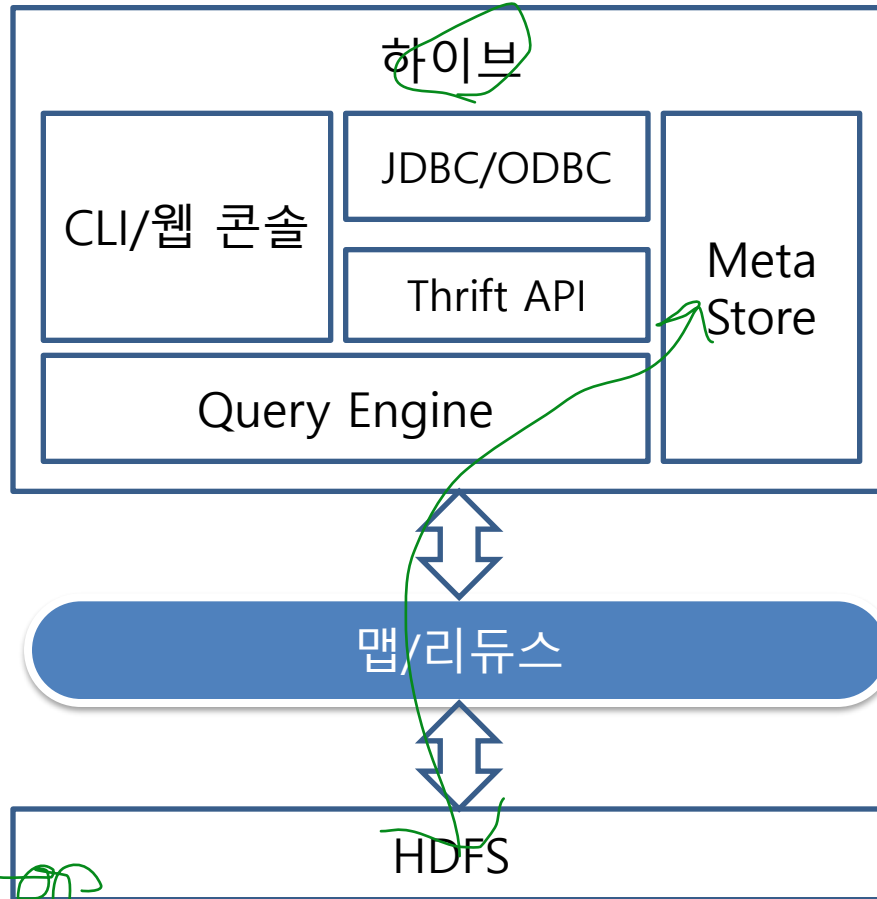
➤ 스크립

- 가공/분석된 데이터를 외부로 제공하며, 필요 시 분석/응용 단계에서도 사용

➤ 우지

- 데이터의 품질을 높이는 단계, 과정이 길고 복잡해지기 마련 .
- 프로세스를 구성해 복잡도를 낮추고, 자동화.

➤ Hive 아키텍처



CLI : 사용자가 Hive 쿼리를 입력 및 실행 인터페이스

JDBC/ODBC Driver : Hive 쿼리를 다양한 DB와 연결하기 위한 드라이버

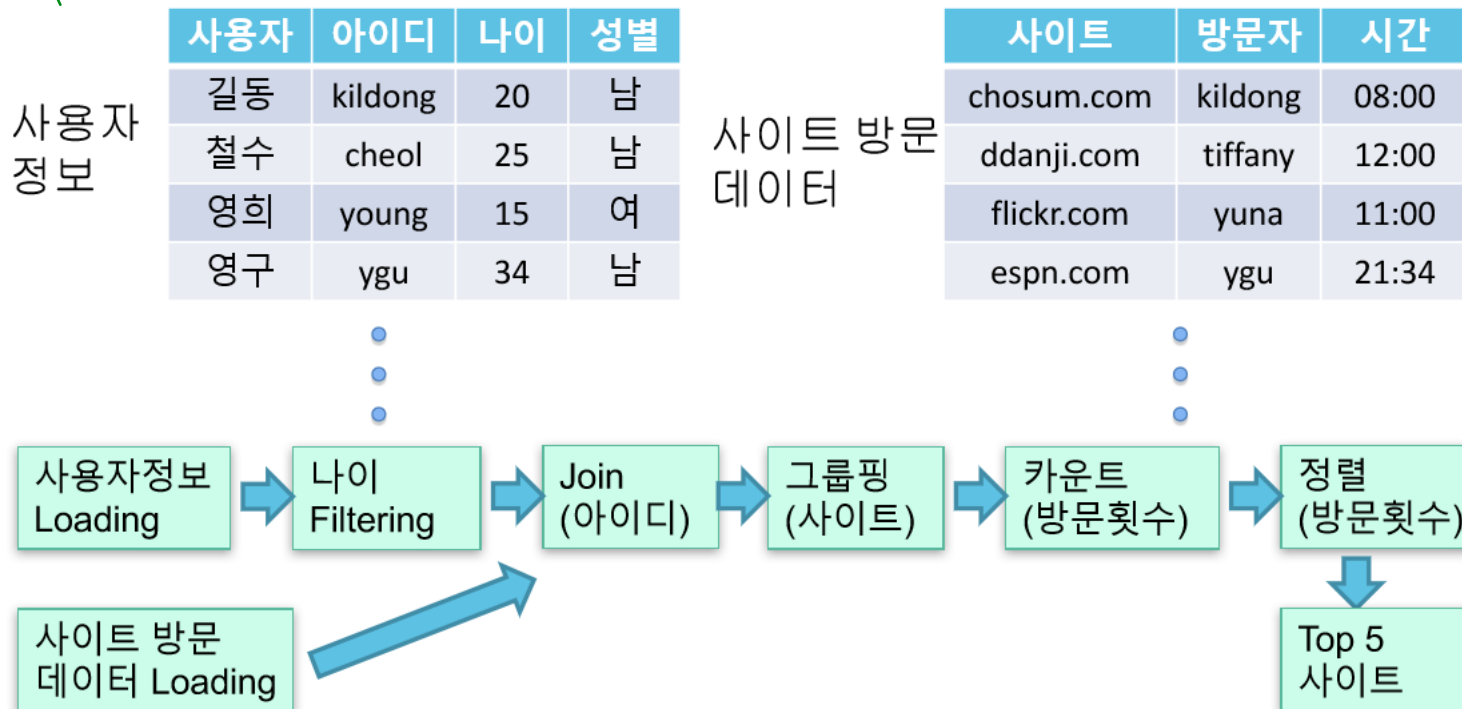
Query Engine : Hive QL을 맵리듀스 코드로 변환/실행

MetaStore : Hive에서 사용하는 테이블의 스키마 정보를 저장 및 관리 기본적으로 Derby DB 사용 (DBMS 변경 가능)

MapReduce vs Hive

문제

- 18 ~ 25세 연령대의 사용자가 가장 많이 방문하는 사이트 5개를 찾아라



12

- 직접 MapReduce 프로그램을 코딩 할 경우

```

        string outval = key + "," + val + ",";
        cc.collect(current, new Next(outval));
        reporter.setState("on");
    }

    }

    public static class Reducemerge extends MapReduceBase
    implements Mapper<Text, Text, Text, LongWritable> {

        public void map(
            Text val
            OutputCollector<Text, Text, Text, LongWritable> cc,
            Reporter reporter) throws IOException {
            // find the val in the string
            String line = val.toString();
            int firstcomma = line.indexOf(',');
            int secondcomma = line.indexOf(',', firstcomma);

            String key = line.substring(0, firstcomma);
            secondcomma++;
            // drop the rest of the record, I don't need it
            anymore;
            // just pass a 1 for the combiner/reducer to
            sum instead.
            Text outkey = new Text(key);
            cc.collect(outkey, new LongWritable(1));
        }

        public static class Reducemerge2 extends MapReduceBase
        implements Reducer<Text, LongWritable,
            WritableAccumulator> {

            public void reduce(
                Text[] keys,
                WritableAccumulator<Text, LongWritable> iter,
                OutputCollector<Text, Text, Text, LongWritable> cc,
                Reporter reporter) throws IOException {
                // Add up all the values we see
                long sum = 0;
                while (iter.hasNext()) {
                    sum += iter.next().get();
                }
                cc.collect(key, new LongWritable(sum));
            }

        }

        public static class Reddoallkeys extends MapReduceBase
        implements Mapper<WritableComparable, Writable,
            LongWritable, Text> {

            public void map(
                WritableComparable key,
                Writable val,
                OutputCollector<Text, LongWritable, Text> cc,
                Reporter reporter) throws IOException {
                cc.collect(key.toString(), val, (Text)key,
                    new LongWritable(1));
            }

        }

        public static class Sumallkeys extends MapReduceBase
        implements Reducer<Text, LongWritable, Text, LongWritable> {

            Text key;

            int count = 0;

            public void reduce(
                Text[] keys,
                WritableAccumulator<Text, LongWritable> iter,
                OutputCollector<Text, LongWritable, Text> cc,
                Reporter reporter) throws IOException {
                // only output the first 100 records
                while (count < 100 && iter.hasNext()) {
                    cc.collect(key, iter.next());
                }
            }

        }

        public static void main(String[] args) throws
            IOException {
            JobConf job = new JobConf(Main.class);
            job.setMapperClass(Reducemerge.class);
            job.setReducerClass(Reducemerge2.class);
            job.setOutputFormatClass(Text.class);
            job.setInputFormatClass(Text.class);
            FileOutputFormat.setOutputPath(job, new
                Path("user/Gates/"));
            job.setOutputFormatClass(Text.class);
            job.setNewClass(Main.class);
            job.setJarInplace(true);
            job.waitForCompletion(true);
        }
    }
}

```

```

ip_addrname=make(0);
loadname = new subcommand();
subcommand ifu = new subcommand(sample_name.class);
join.++subcommand=ifu;
ifu.++inputformat=chaininputformat.class;
ifu.++outputformat=chainoutputformat.class;
ifu.++inputname=chaininputname.class;
ifu.++outputname=chainoutputname.class;
ifu.++groupname=chaingroupname.class;
ifu.++inputpathformat=chaininputpathformat.class;
ifu.++outputpathformat=chainoutputpathformat.class;
ifu.++inputpath=chaininputpath(ifu, new
fileoutputformat().getoutputpath(ifu, new
new path("/user/gates/cmp/filtered_users"));
ifu.++reduce=make(0);
ifu.++loadname=ifu;

subcommand join = new subcommand(sample_name.class);
join.++subcommand=join_users_and_wages;
join.++inputformat=chaininputformat.class;
join.++outputformat=chainoutputformat.class;
join.++inputname=chaininputname.class;
join.++outputname=chainoutputname.class;
join.++groupname=chaingroupname.class;
join.++inputpathformat=chaininputpathformat.class;
join.++outputpathformat=chainoutputpathformat.class;
join.++inputpath=chaininputpath(join, new
path("/user/gates/cmp/indexed_pages"));
join.++reduce=chainreduce(chain_users_and_wages, new
path("/user/gates/cmp/filtered_users"));
join.++loadname=chainload(chain_users_and_wages, new
path("/user/gates/cmp/joined"));
join.++groupname=chaingroup;
join.joinIn = new job(join);
chainJob.addPendingIngress(chainUsers);
chainJob.addPendingIngress(chainUsers);

subcommand group = new subcommand(sample_name.class);
group.++subcommand=ifu;
group.++inputformat=newvalinputformat.class;
group.++outputformat=chainoutputformat.class;
group.++inputname=chaininputname.class;
group.++outputname=chainoutputname.class;
group.++groupname=chaingroupname.class;
group.++inputpathformat=chaininputpathformat.class;
group.++outputpathformat=chainoutputpathformat.class;
group.++inputpath=chaininputpath(group, new
path("/user/gates/cmp/filtered_users"));
group.++reduce=chainreduce(chainUsers, new
path("/user/gates/cmp/joined"));
group.groupIn = new job(group);
groupJob.addPendingIngress(chainUsers);

subcommand top10 = new subcommand(sample_name.class);
top10.++subcommand=ifu;
top10.++inputformat=chaininputformat.class;
top10.++outputformat=chainoutputformat.class;
top10.++inputname=chaininputname.class;
top10.++outputname=chainoutputname.class;
top10.++groupname=chaingroupname.class;
top10.++inputpathformat=chaininputpathformat.class;
top10.++outputpathformat=chainoutputpathformat.class;
top10.++inputpath=chaininputpath(top10, new
path("/user/gates/cmp/filtered_users"));
top10.++reduce=chainreduce(chainUsers, new
path("/user/gates/cmp/joined"));
top10.++loadname=chainload(chainUsers, new
path("/user/gates/cmp/joined"));
top10.++groupname=chaingroup;

for user in chaincommand["find top 100 lists"]
do
  1c.addJob(loadname);
  1c.addJob(reduce);
  1c.addJob(chainUsers);
  1c.addJob(chaingroup);
  1c.addJob(chainload);
  1c.run();

```

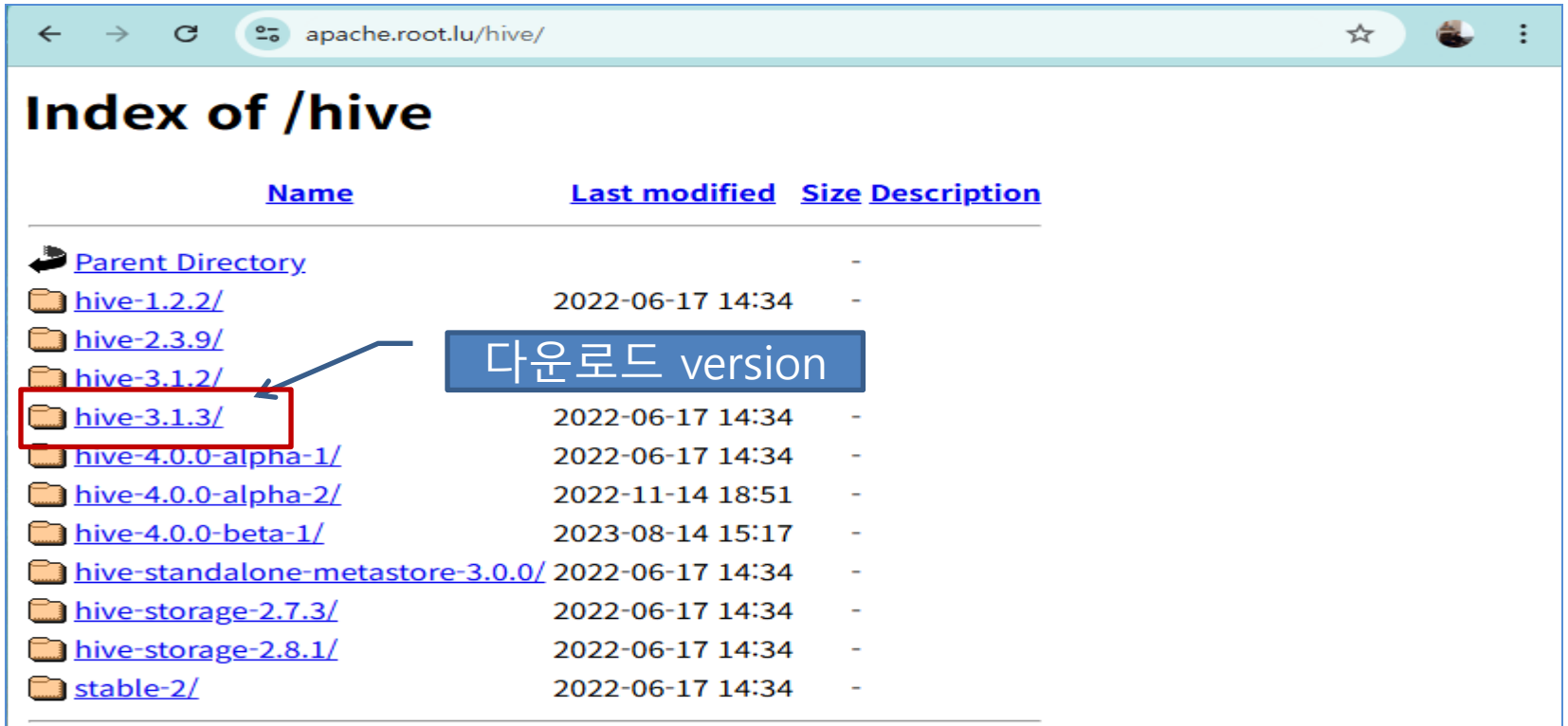
Hive SQL

사용자 중심 편의성 제공, 기본 SQL문 이용 빅데이터 검색, 시간과 노력 절감, 간단한 쿼리문

```
select  t2.url,      count(1) as      visits      from      userinfo t1
        join      visitinfo t2      on      (t1.id=t2.id)      where
        t1.age    >      17      and      t1.age    <      26
        group    by      t2.url      sort      by      visits      DESC
        limit    5;
```


2. Hive 설치

<https://apache.root.lu/hive/>에서 다운로드 가능한 hive 버전 확인



The screenshot shows the Apache Hive website's index page. The browser address bar displays <https://apache.root.lu/hive/>. The page title is "Index of /hive". Below the title is a table with columns: Name, Last modified, Size, and Description. The table lists various Hive versions and components. The entry for [hive-3.1.3/](#) is highlighted with a red box, and a blue arrow points to it from a callout box labeled "다운로드 version".

Name	Last modified	Size	Description
Parent Directory	-	-	-
hive-1.2.2/	2022-06-17 14:34	-	-
hive-2.3.9/	-	-	-
hive-3.1.2/	-	-	-
hive-3.1.3/	2022-06-17 14:34	-	-
hive-4.0.0-alpha-1/	2022-06-17 14:34	-	-
hive-4.0.0-alpha-2/	2022-11-14 18:51	-	-
hive-4.0.0-beta-1/	2023-08-14 15:17	-	-
hive-standalone-metastore-3.0.0/	2022-06-17 14:34	-	-
hive-storage-2.7.3/	2022-06-17 14:34	-	-
hive-storage-2.8.1/	2022-06-17 14:34	-	-
stable-2/	2022-06-17 14:34	-	-

<https://apache.root.lu/hive/hive-3.1.3/> 에서 hive 압축파일 다운로드



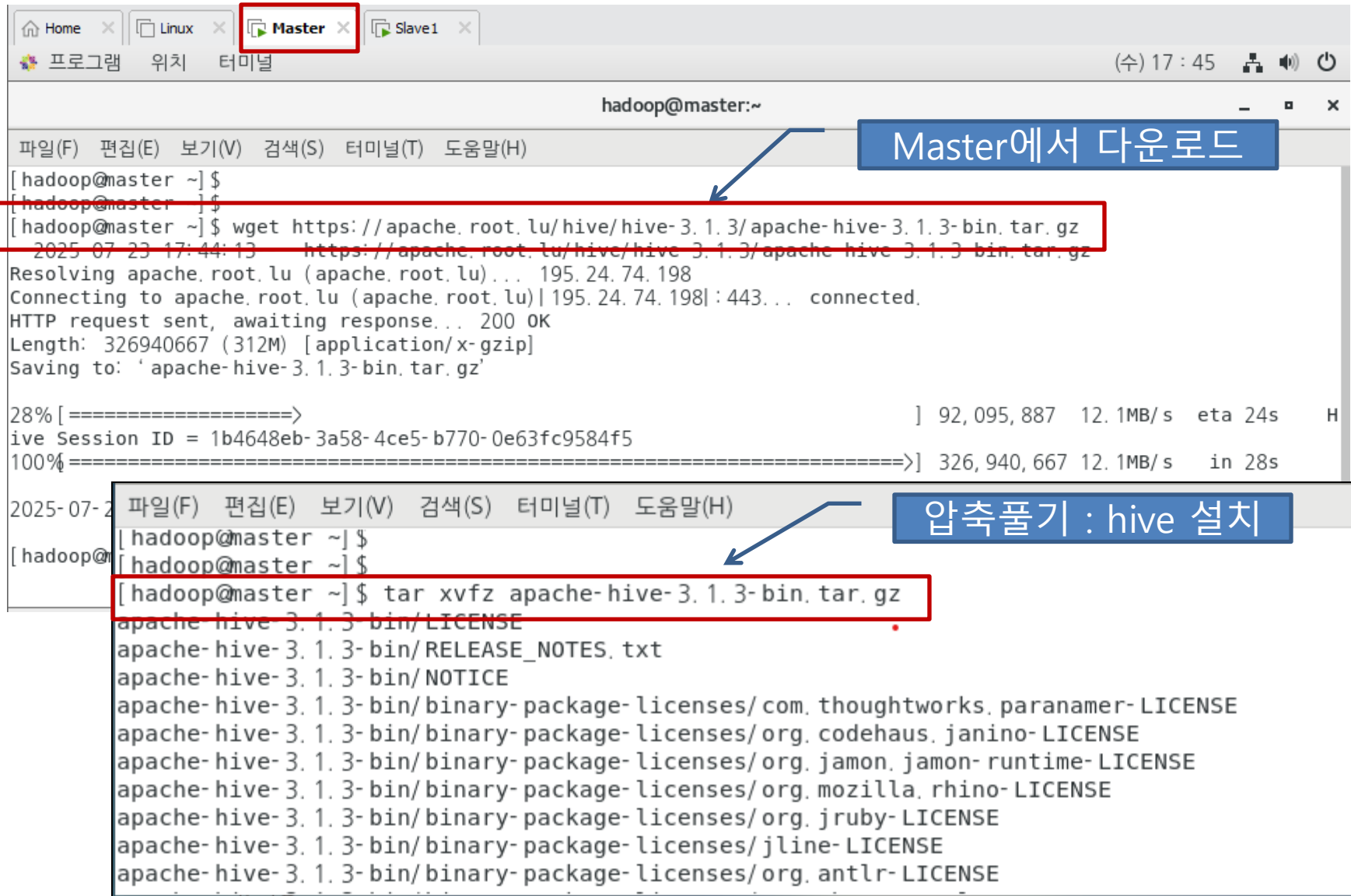
← → ↻ apache.root.lu/hive/hive-3.1.3/ ☆ 👤 ⋮

Index of /hive/hive-3.1.3

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory			
 apache-hive-3.1.3-bin.tar.gz	2022-04-08 19:42	312M	
 apache-hive-3.1.3-src.tar.gz	2022-04-08 19:42	25M	

Apache/2.4.62 (Debian) Server at apache.root.lu Port 443

1) Hive 다운로드 & 압축풀기(Master 작업)



The screenshot shows a terminal window with two tabs: 'Master' and 'Slave1'. The 'Master' tab is active. The terminal prompt is 'hadoop@master:~'. A blue box with the text 'Master에서 다운로드' (Download on Master) has an arrow pointing to the 'wget' command. The command is 'wget https://apache.root.lu/hive/hive-3.1.3/apache-hive-3.1.3-bin.tar.gz', which is highlighted with a red box. The output shows the file being downloaded from 'https://apache.root.lu/hive/hive-3.1.3/apache-hive-3.1.3-bin.tar.gz' with a length of 326,940,667 bytes (312M) and a download time of 28 seconds. A second blue box with the text '압축풀기 : hive 설치' (Unzip : hive installation) has an arrow pointing to the 'tar xvfz' command. The command is 'tar xvfz apache-hive-3.1.3-bin.tar.gz', which is also highlighted with a red box. The output shows the files being extracted, including 'LICENSE', 'RELEASE_NOTES.txt', 'NOTICE', and various license files for different components like 'com.thoughtworks.paranamer', 'org.codehaus.janino', 'org.jamon.jamon-runtime', 'org.mozilla.rhino', 'org.jruby', and 'organtlr'.

```
hadoop@master:~  
[hadoop@master ~]$  
[hadoop@master ~]$  
[hadoop@master ~]$ wget https://apache.root.lu/hive/hive-3.1.3/apache-hive-3.1.3-bin.tar.gz  
2025-07-23 17:44:13 https://apache.root.lu/hive/hive-3.1.3/apache-hive-3.1.3-bin.tar.gz  
Resolving apache.root.lu (apache.root.lu)... 195.24.74.198  
Connecting to apache.root.lu (apache.root.lu)|195.24.74.198|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 326940667 (312M) [application/x-gzip]  
Saving to: 'apache-hive-3.1.3-bin.tar.gz'  
  
28%[=====>] 92,095,887 12.1MB/s eta 24s H  
hive Session ID = 1b4648eb-3a58-4ce5-b770-0e63fc9584f5  
100%[=====>] 326,940,667 12.1MB/s in 28s  
  
2025-07-23 17:44:41 [hadoop@master ~]$  
[hadoop@master ~]$  
[hadoop@master ~]$ tar xvfz apache-hive-3.1.3-bin.tar.gz  
apache-hive-3.1.3-bin/LICENSE  
apache-hive-3.1.3-bin/RELEASE_NOTES.txt  
apache-hive-3.1.3-bin/NOTICE  
apache-hive-3.1.3-bin/binary-package-licenses/com.thoughtworks.paranamer-LICENSE  
apache-hive-3.1.3-bin/binary-package-licenses/org.codehaus.janino-LICENSE  
apache-hive-3.1.3-bin/binary-package-licenses/org.jamon.jamon-runtime-LICENSE  
apache-hive-3.1.3-bin/binary-package-licenses/org.mozilla.rhino-LICENSE  
apache-hive-3.1.3-bin/binary-package-licenses/org.jruby-LICENSE  
apache-hive-3.1.3-bin/binary-package-licenses/jline-LICENSE  
apache-hive-3.1.3-bin/binary-package-licenses/org.antlr-LICENSE
```

2) Hive 소프트웨어 링크 적용

```
프로그램  위치  터미널  ko (금) 15:56
```

```
hadoop@master:~
```

파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

```
[hadoop@master ~]$  
[hadoop@master ~]$ ln -s apache-hive-3.1.3-bin hive  
[hadoop@master ~]$  
[hadoop@master ~]$ ls -l
```

합계 1175792

drwxrwxr-x.	10	hadoop	hadoop	4096	7월	5	15:49	apache-hive-3.1.3-bin
-rw-rw-r--.	1	hadoop	hadoop	326940667	4월	9	2022	apache-hive-3.1.3-bin
drwxr-xr-x.	11	hadoop	hadoop	4096	7월	5	11:06	hadoop-3.3.6
-rw-rw-r--.	1	hadoop	hadoop	730107476	6월	26	2023	hadoop-3.3.6.tar.gz
drwxrwxr-x.	3	hadoop	hadoop	4096	7월	5	10:53	hadoopdata
lrwxrwxrwx.	1	hadoop	hadoop	21	7월	5	15:56	hive -> apache-hive-3.1.3-bin
-rw-rw-r--.	1	hadoop	hadoop	146902735	7월	4	12:07	jdk-8u411-linux-x64.tar.gz
drwxr-xr-x.	2	hadoop	hadoop	4096	7월	4	11:34	고개

Soft link 적용

hive 별칭 사용

3) Hive 홈 PATH설정

- Hive 디렉터리 이동 및 내용 보기

The screenshot shows a terminal window with the title bar 'hadoop@master:~/hive'. The terminal output is as follows:

```
[hadoop@master ~]$  
[hadoop@master ~]$ cd hive  
[hadoop@master hive]$  
[hadoop@master hive]$ pwd  
/home/hadoop/hive  
[hadoop@master hive]$  
[hadoop@master hive]$ ls  
LICENSE  RELEASE_NOTES.txt  binary-package-licenses  examples  jdbc  scripts  
NOTICE   bin                 conf                     hcatalog  lib
```

Annotations on the image:

- A blue box labeled 'Hive 홈 디렉터리 이동' points to the command `cd hive`.
- A blue box labeled 'Hive 경로 확인' points to the command `pwd`.
- A blue box labeled 'Hive 실행파일/환경설정 파일' points to the `ls` command and its output.
- Red boxes highlight the commands `cd hive`, `pwd`, and `ls`.
- Purple boxes highlight the output files `bin` and `conf` in the `ls` output.

- ❖ bin : hive 실행 파일 디렉터리
- ❖ conf : hive 환경설정 파일 디렉터리

.bash_profile 설정

```
[hadoop@master hive]$  
[hadoop@master hive]$  
[hadoop@master hive]$ cd  
[hadoop@master ~]$  
[hadoop@master ~]$ vi .bash_profile
```

홈 디렉터리 이동

설정파일 열기

```
hadoop@master:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
  
# User specific environment and startup programs  
  
PATH=$PATH: $HOME/.local/bin: $HOME/bin  
  
export PATH  
  
export JAVA_HOME=/usr/local/jdk1.8.0_461  
export HADOOP_HOME=/home/hadoop/hadoop-3.3.6  
export PATH=$PATH: $JAVA_HOME/bin: $HADOOP_HOME/bin: $HADOOP_HOME/sbin  
  
# hive  
export HIVE_HOME=/home/hadoop/hive  
export PATH=$PATH: $HIVE_HOME/bin
```

Hive 환경변수

Hive PATH 추가

20, 1 바닥

.bash_profile 적용/테스트

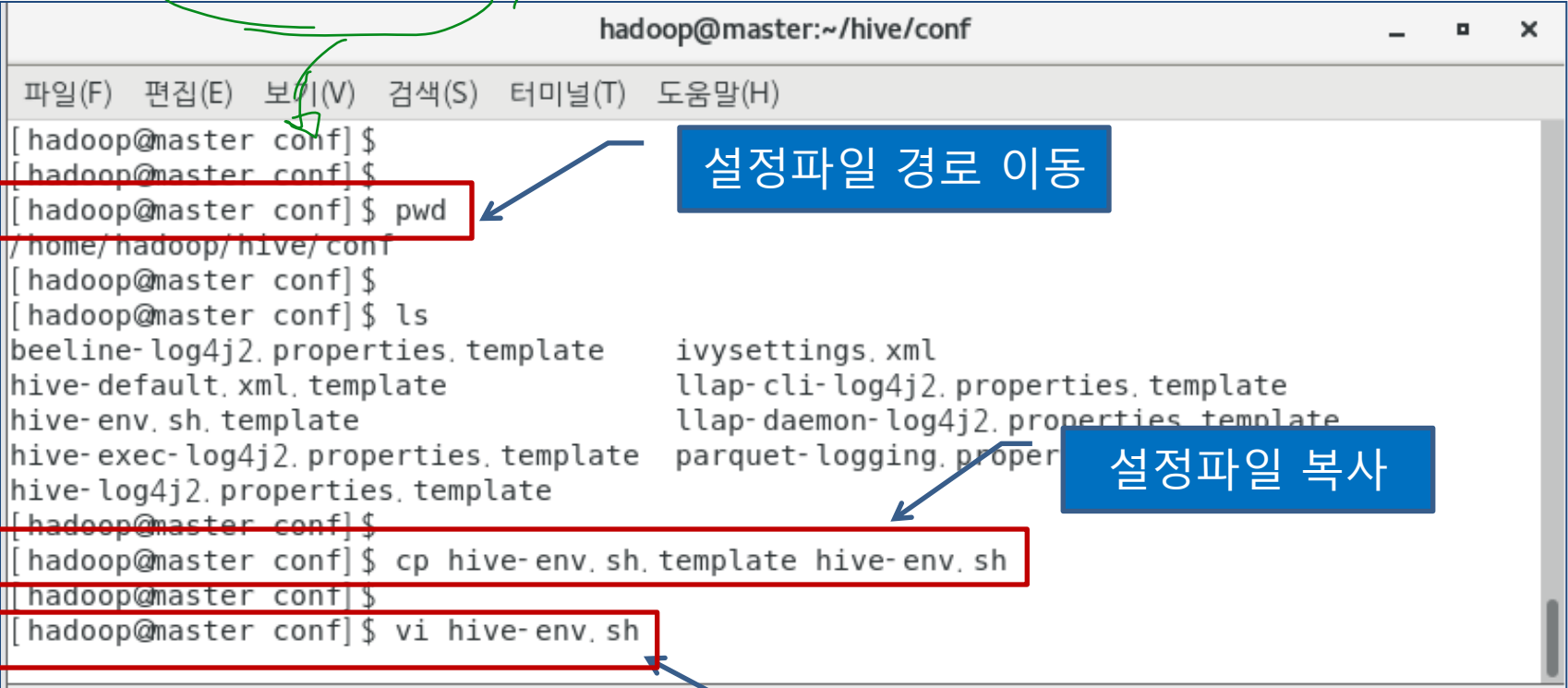
```
hadoop@master:~/hive
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[hadoop@master ~]$
[hadoop@master ~]$
[hadoop@master ~]$ source .bash_profile
[hadoop@master ~]$
[hadoop@master ~]$ cd $HIVE_HOME
[hadoop@master hive]$
[hadoop@master hive]$ pwd
/home/hadoop/hive
[hadoop@master hive]$ echo $PATH
/usr/local/jdk1.8.0_411/bin: /usr/local/bin: /usr/bin: /usr/sbin: /bin: /sbin: /home/hadoop/.local/bin: /home/hadoop/bin: /home/hadoop/hadoop-3.3.6/bin: /home/hadoop/hadoop-3.3.6/sbin: /home/hadoop/.local/bin: /home/hadoop/bin: /usr/local/jdk1.8.0_411/bin: /home/hadoop/hadoop-3.3.6/bin: /home/hadoop/hadoop-3.3.6/sbin: /home/hadoop/hive/bin
[hadoop@master hive]$
```

3. Hive 환경설정

- 1) hive-env.sh : 환경변수 설정
- 2) hive-site.xml : Hive의 핵심 동작 설정
- 3) hive-config.sh : HDFS와 Hive의 상호작용을 위한 환경 설정

1) Hive 환경/실행파일 경로설정

하기전에 cd conf!!!



The image shows a terminal window titled 'hadoop@master:~/hive/conf'. The terminal output is as follows:

```
[hadoop@master conf]$  
[hadoop@master conf]$  
[hadoop@master conf]$ pwd  
/home/hadoop/hive/conf  
[hadoop@master conf]$  
[hadoop@master conf]$ ls  
beeline-log4j2.properties.template      ivysettings.xml  
hive-default.xml.template                llap-cli-log4j2.properties.template  
hive-env.sh.template                    llap-daemon-log4j2.properties.template  
hive-exec-log4j2.properties.template    parquet-logging.properties  
hive-log4j2.properties.template  
[hadoop@master conf]$  
[hadoop@master conf]$ cp hive-env.sh.template hive-env.sh  
[hadoop@master conf]$  
[hadoop@master conf]$ vi hive-env.sh
```

Annotations in the image include:

- A green circle around the text '하기전에 cd conf!!!' with an arrow pointing to the terminal prompt.
- A blue box labeled '설정파일 경로 이동' (Move configuration file path) with an arrow pointing to the 'pwd' command.
- A blue box labeled '설정파일 복사' (Copy configuration file) with an arrow pointing to the 'cp' command.
- A blue box labeled '설정파일 열기' (Open configuration file) with an arrow pointing to the 'vi' command.

```
hadoop@master:~/hive/conf

파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

# export HADOOP_HEAPSIZE=1024
#
# Larger heap size may be required when running queries over large number of files or par
titions.
# By default hive shell scripts use a heap size of 256 (MB). Larger heap size would also
be
# appropriate for hive server.

# Set HADOOP_HOME to point to a specific hadoop install directory
# HADOOP_HOME=${ bin} /.../hadoop

# Hive Configuration Directory can be controlled by:
# export HIVE_CONF_DIR=

# Folder containing extra libraries required for
lled by:
# export HIVE_AUX_JARS_PATH=

export HIVE_CONF_DIR=/home/hadoop/hive/conf
export HIVE_AUX_JARS_PATH=/home/hadoop/hive/lib

: wq
```

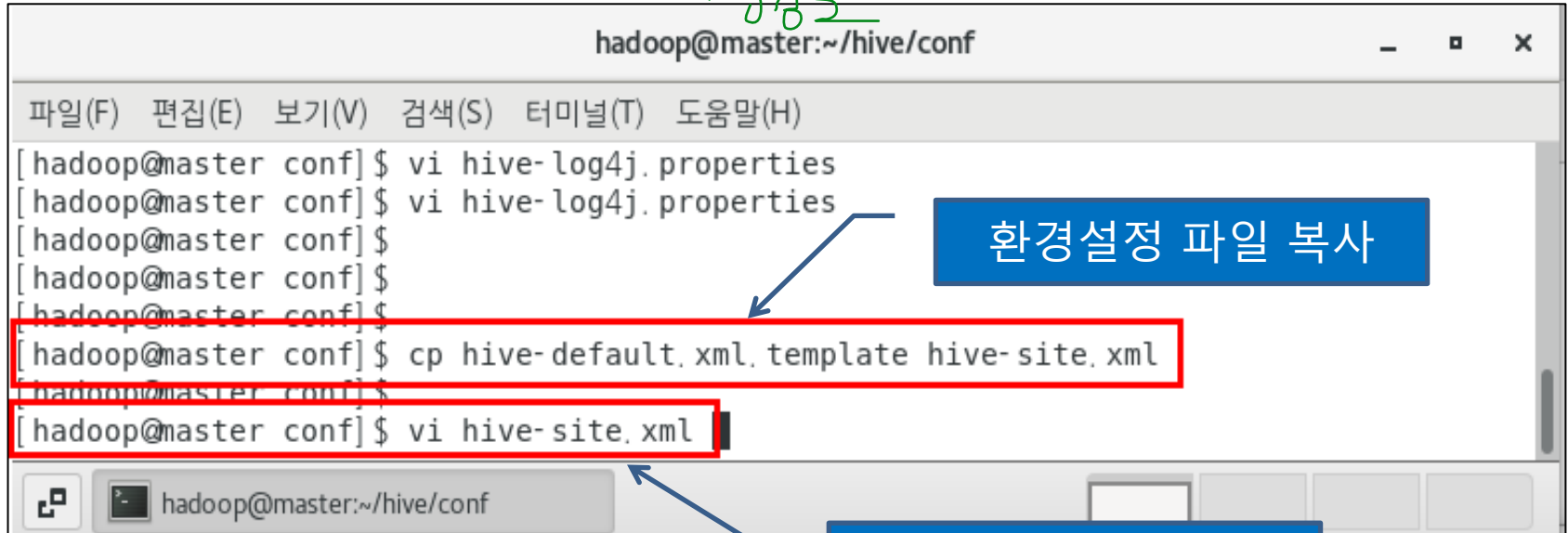
Hive 환경설정, jar파일 경로

60, 3

바닥

2) hive-site.xml 환경 설정 : metastore & 사용자 설정

저장 경로



A terminal window titled 'hadoop@master:~/hive/conf' with a menu bar (파일(F), 편집(E), 보기(V), 검색(S), 터미널(T), 도움말(H)). The terminal shows the following commands:

```
[hadoop@master conf]$ vi hive-log4j.properties
[hadoop@master conf]$ vi hive-log4j.properties
[hadoop@master conf]$
[hadoop@master conf]$
[hadoop@master conf]$
[hadoop@master conf]$ cp hive-default.xml.template hive-site.xml
[hadoop@master conf]$
[hadoop@master conf]$ vi hive-site.xml
```

The last two commands are highlighted with red boxes. A blue arrow points from the first red box to a blue callout box, and another blue arrow points from the second red box to a blue callout box.

환경설정 파일 복사

환경설정 파일 열기

- metastore 경로 확인 : hive table 저장 위치

```
/nsec), which is day if not specified.  
Time-To-Live (TTL) of proto event files before cleanup.  
</description>  
</property>  
<property>  
  <name>hive.hook.proto.file.per.event</name>  
  <value>>false</value>  
@  
/metastore.warehouse
```

문자열 검색

슬래시 하면 바로 타이핑 가능 /metastore.warehouse하고 엔터하면 찾아줌-> 그중 위에꺼

hadoop@master:~/hive/conf

파일(F) 편집(E) 보기(V) 검색(S) 터미널

```
</property>  
<property>  
  <name>hive.metastore.warehouse.dir</name>  
  <value>user/hive/warehouse</value>  
</description>location of default database for the warehouse</description>  
</property>  
<property>
```

metastore 경로 확인(Hive table 저장 위치)

위치 맞는지 확인

334, 16

8%

- java관련 임시 디렉터리와 hive 사용자 이름 설정

```
hadoop@master:~/hive/conf
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
<value>0</value>
<description>
  Expects value bigger than 0.
  If value is greater than 0 logs in fixed intervals of size n rather than exponential
ly.
</description>
</property>
<property>
  <name>system: java. io. tmpdir</name>
  <value>/tmp/hive/java</value>
</property>
<property>
  <name>system: user. name</name>
  <value>${user.name}</value>
</property>
</configuration>
~
: wq
```

파일 끝 부분의
</configuration> 태그 안쪽에 추가
파일 끝 커서 이동 : 단축키(G)

SHIFT+g하면 맨말로-> /configuration위에 두개의 property삽입

3) hive-config.sh 환경 설정 : HDFS와 hive의 상호작용

```
hadoop@master:~/hive/bin
[hadop@master conf]$
[hadop@master conf]$ cd $HIVE_HOME/bin
[hadop@master bin]$
[hadop@master bin]$ ls
beeline  hive  hiveserver2  init-hive-dfs.sh  replstate.sh
ext  hive-config.sh  hplsql  metastool
[hadop@master bin]$
[hadop@master bin]$ vi hive-config.sh

export HIVE_CONF_DIR=$HIVE_CONF_DIR
export HIVE_AUX_JARS_PATH=$HIVE_AUX_JARS_PATH

# Default to use 256MB
export HADOOP_HEAPSIZE=${HADOOP_HEAPSIZE:-256}

# Disable the JNDI. This feature has critical RCE v
# when 2. x <= log4j.version <= 2.14.1
export HADOOP_CLIENT_OPTS="$HADOOP_CLIENT_OPTS -Dlog4j2.formatMsgNoLookups=true"

export HADOOP_HOME=/home/hadoop/hadoop-3.3.6
```

1. bin 디렉터리 이동

2. 환경설정 파일 열기

3. 파일 끝 부분에 Hadoop 홈 디렉터리 추가

4) javax.jdo 수정

Hive 어플리케이션이 초기 설정 단계에서 데이터베이스가 존재하지 않을 경우를 대비하여 새로운 데이터베이스를 생성하는 환경설정 부분이다.

1 vi \$HIVE_HOME/conf/hive-site.xml

1. 환경설정 파일 열기

2 /javax.jdo.option.ConnectionURL

2. 태그 찾기

```
hadoop@master:~/hive/conf
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
<property>
  <name>javax.jdo.option.Multithreaded</name>
  <value>true</value>
  <description>Set this to true if multiple threads access metastore
</property>
<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:derby:/home/hadoop/apache-hive-3.1.3-bin/metastore_db;create=true</value>
  <description>
    JDBC connect string for a JDBC metastore.
    To use SSL to encrypt/authenticate the connection, provide database-specific SSL flag in the connection URL.
587, 104 8%
hadoop@master:~/hive/conf [hadoop@master:~]
```

이거 새로이 삽입

5) derby database 시작 : HIVE에서 사용할 DB 초기화

오류 발생 : hive-site.xml의 3215행에 **이상한 문자** 발견

```
hadoop@master:~/hive/conf
[hadop@master conf]$ schematool -initSchema -dbType derby
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop-3.3.6/share/hadoop/common/lib/log4j-slf4j-impl-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Exception in thread "main" java.lang.RuntimeException: com.ctc.wstx.exc.WstxUnexpectedChar: Illegal character entity: expansion character (code 0x8)
at [row,col,system-id]: [3215,96,"file:/home/hadoop/apache-hive-3.1.3-bin/conf/hive-site.xml"]
    at org.apache.hadoop.conf.Configuration.loadResource(Configuration.java:3101)
    at org.apache.hadoop.conf.Configuration.loadResources(Configuration.java:3050)
    at org.apache.hadoop.conf.Configuration.loadProps(Configuration.java:2923)
    at org.apache.hadoop.conf.Configuration.addResourceObject(Configuration.java:1035)
    at org.apache.hadoop.conf.Configuration.addResource(Configuration.java:940)
    at org.apache.hadoop.hive.conf.HiveConf.initialize(HiveConf.java:5154)
    at org.apache.hadoop.hive.conf.HiveConf.<init>(HiveConf.java:5107)
    at org.apache.hadoop.hive.beeline.HiveSchemaTool.<init>(HiveSchemaTool.java:96)
    at org.apache.hadoop.hive.beeline.HiveSchemaTool.main(HiveSchemaTool.java:1473)
```

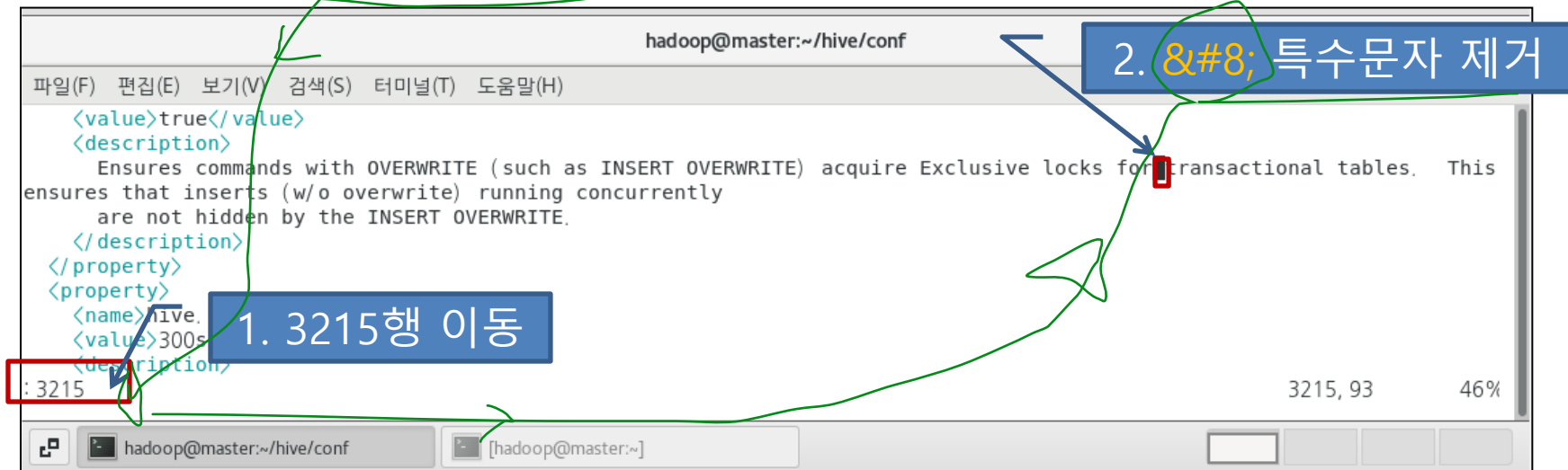
1. derby초기화

그러면 오류 발생

Hive 실행 시 오류 발생

오류 원인 : hive-site.xml 파일의 3215행 96열 부근의 내용을 확인하세요.
해당 위치에 잘못된 문자(코드 0x8)가 있는지 확인

vi \$HIVE_HOME/conf/hive-site.xml



6) derby database 재시작

아래 명령어를 실행하면 apache-hive-3.1.3-bin/metastore_db 생성됨

The image displays two terminal windows from a VMware Workstation 16 Player. The top window, titled 'hadoop@master:~/hive/conf', shows the execution of the command `$HIVE_HOME/bin/schematool -initSchema -dbType derby` to initialize the Derby database. A blue box labeled '1. derby초기화' points to this command. The output shows the successful creation of the metastore database. A green arrow points from the text 'hive 홈 디렉토리에 metastore_db가 생성' to the `metastore_db` entry in the directory listing of the bottom window. The bottom window, titled 'hadoop@master:~', shows the command `ls $HIVE_HOME/` being executed, with a blue box labeled '2. Hive 홈 디렉터리 보기' pointing to it. The output lists various files and directories, including `metastore_db`, which is highlighted with a red box.

```
hadoop@master:~/hive/conf
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[hadoop@master conf]$ vi hive-site.xml
[hadoop@master conf]$
[hadoop@master conf]$ $HIVE_HOME/bin/schematool -initSchema -dbType derby
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar: file:/home/hadoop/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar: file:/home/hadoop/hadoop-3.3.6/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:      jdbc:derby:/home/hadoop/apache-hive-3.1.3-bin/metastore_db; databaseName=metastore_db; create=true
Metastore Connection Driver :  org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:    APP
Starting metastore schema initialization to 3.1.0
Initial
```

```
hadoop@master:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[hadoop@master ~]$
[hadoop@master ~]$ ls $HIVE_HOME/
LICENSE  RELEASE_NOTES.txt  binary-package-licenses  examples  jdbc  metastore_db
NOTICE   bin                conf                     hcatalog  lib   scripts
[hadoop@master ~]$
```

4. Hadoop DFS 작업

준비 : master/slave1 서버 ON

1) **start-all.sh** # 하둡/얀 실행

```
hadoop@master ~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [master]
master: starting namenode, logging to /home/hadoop/hadoop-2.10.1/logs/hadoop-hadoop-namenode-master.out
slave1: datanode running as process 1807. Stop it first.
Starting secondary namenodes [slave1]
slave1: secondarynamenode running as process 1887. Stop it first.
starting yarn daemons
starting resourcemanager, logging to /home/hadoop/hadoop-2.10.1/logs/yarn-hadoop-resourcemanager-master.out
hadoop@master ~$ jps
5096 Jps
3944 ResourceManager
3450 NameNode
3694 SecondaryNameNode
hadoop@master ~$
```

hive는 HDFS와 YARN 사용

Master 서버에서 동작 중인 대몬

2) 작업 디렉터리 생성/접근권한 설정

```
hadoop@master:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[hadoop@master ~]$  
[hadoop@master ~]$  
[hadoop@master ~]$  
[hadoop@master ~]$ hdfs dfs -ls /  
Found 3 items  
drwxr-xr-x - hadoop supergroup 0 2021-12-31 12:12 /output  
drwxr-xr-x - hadoop supergroup 0 2021-12-31 12:07 /test  
drwxr-xr-x - hadoop supergroup 0 2022-01-03 14:56 /tmp  
[hadoop@master ~]$  
[hadoop@master ~]$  
[hadoop@master ~]$ hdfs dfs -rm -R /tmp  
Deleted /tmp  
[hadoop@master ~]$  
[hadoop@master ~]$  
[hadoop@master ~]$ hdfs dfs -mkdir /tmp  
[hadoop@master ~]$  
[hadoop@master ~]$ hdfs dfs -mkdir -p /user/hive/warehouse  
[hadoop@master ~]$  
[hadoop@master ~]$ hdfs dfs -chmod g+w /tmp  
[hadoop@master ~]$  
[hadoop@master ~]$ hdfs dfs -chmod g+w /user/hive/warehouse  
[hadoop@master ~]$  
[hadoop@master ~]$
```

HDFS 디렉터리 보기

tmp 디렉터리 확인

있는 경우 삭제

tmp 디렉터리 삭제

임시디렉터리 생성

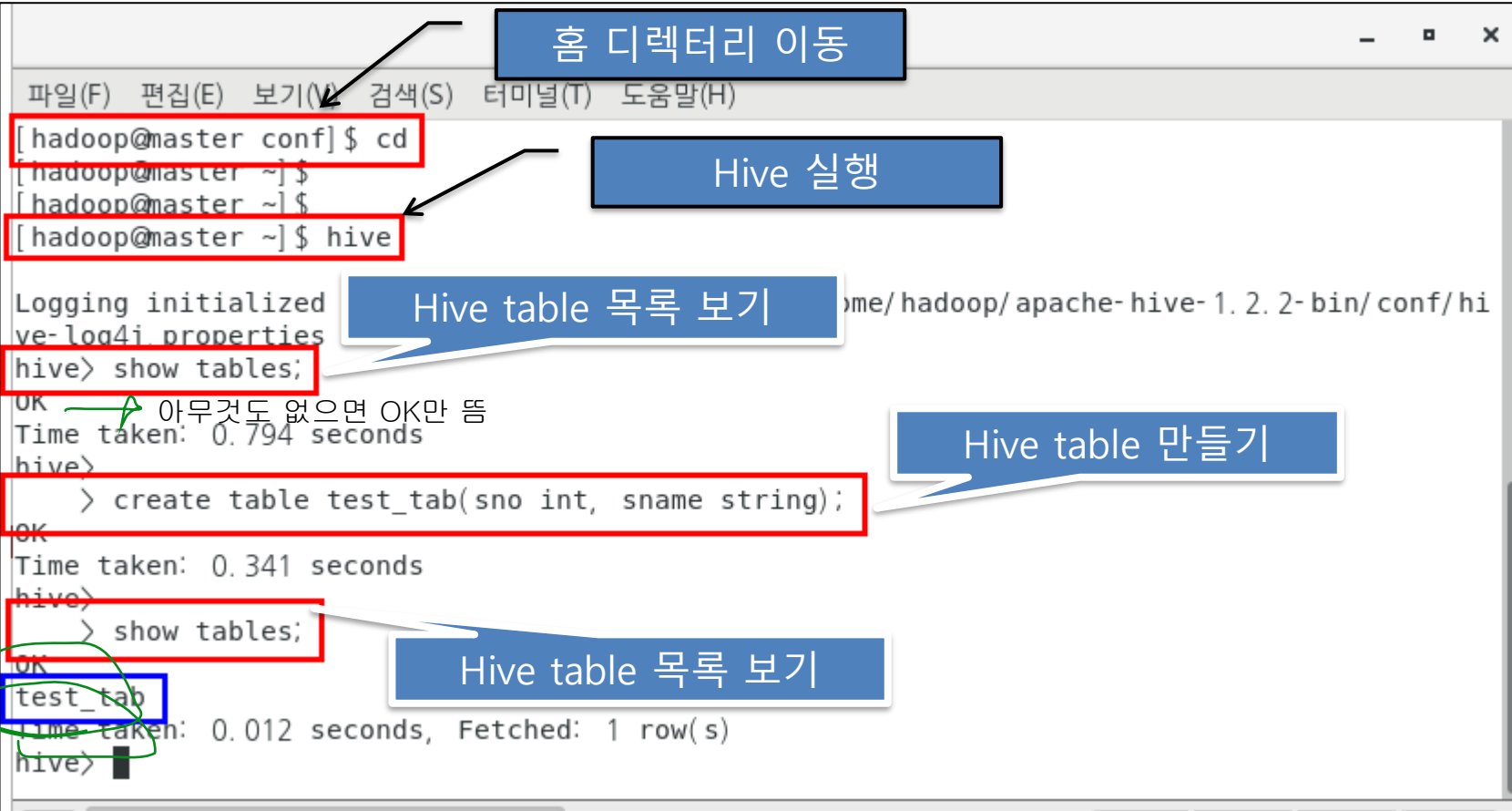
tmp 디렉터리 생성

hive table 저장 디렉터리 생성

접근권한 변경

5. Hive 실행 & 테이블 만들기

1) 사용자 홈 디렉터리 이동 & hive 실행 및 테이블 만들기



A terminal window showing the steps to start Hive and create a table. The window has a menu bar with '파일(F)', '편집(E)', '보기(V)', '검색(S)', '터미널(T)', and '도움말(H)'. The terminal output is as follows:

```
[hadoop@master conf]$ cd
[hadoop@master ~]$
[hadoop@master ~]$ hive

Logging initialized
hive-log4j.properties
hive> show tables;
OK
Time taken: 0.794 seconds
hive>
> create table test_tab(sno int, sname string);
OK
Time taken: 0.341 seconds
hive>
> show tables;
OK
test_tab
Time taken: 0.012 seconds, Fetched: 1 row(s)
hive>
```

Annotations in the image:

- 홈 디렉터리 이동**: Points to the `cd` command.
- Hive 실행**: Points to the `hive` command.
- Hive table 목록 보기**: Points to the first `show tables;` command.
- Hive table 만들기**: Points to the `create table test_tab(sno int, sname string);` command.
- Hive table 목록 보기**: Points to the second `show tables;` command.

Additional notes in the image:

- A green arrow points to the `OK` output of the first `show tables;` command, with the text "아무것도 없으면 OK만 뜸" (If nothing is there, only OK will appear).
- The output `test_tab` in the second `show tables;` command is circled in blue.

2) Hive 테이블 만들기

```
hadoop@master:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
pl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop-3.3.6/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.  
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]  
Hive Session ID = 1fd95dc6-fae0-4c44-b2f8-ec7a910ce658  
  
Logging initialized using configuration in jar:file:/home/hadoop/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j.properties  
Hive Session ID = 2a2...  
Hive-on-MR is deprecated and will be removed in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.  
hive> show tables;  
OK  
Time taken: 0.541 seconds  
hive>  
> create table test_tab(sno int, sname string);  
OK  
Time taken: 0.655 seconds  
hive>  
> show tables;  
OK  
test_tab  
Time taken: 0.029 seconds, fetched: 1 row(s)  
hive>
```

Hive table 목록 보기

Hive table 만들기

Hive table 목록 보기

HIVE 테이블

3) Hive 종료 & HDFS의 디렉터리에서 테이블 확인

```
hive>  
  > show tables;  
OK  
test_tab  
Time taken: 0.029 s  
hive> quit;  
[hadoop@master ~]$  
[hadoop@master ~]$  
[hadoop@master ~]$  
[hadoop@master ~]$ hdfs dfs -ls /user/hive/warehouse  
Found 1 items  
drwxr-xr-x  - hadoop supergroup          0 2025-07-24 14:50 /user/hive/warehouse/test_tab  
[hadoop@master ~]$  
[hadoop@master ~]$
```

Hive 종료

HDFS에서 테이블 확인

HIVE 테이블