**Sqoop1**

**Sqoop1과 Sqoop2의 차이**

|  |  |
| --- | --- |
| **Sqoop1** | **Sqoop2** |
| 1. Client-side install  2. JDBC Driver는 접속하는 Local마다 설치 필요  3. CLI(Command Line Interface) 제공 | 1. Server-side install  2. JDBC Driver가 한 곳만 설치  3. CLI 접속 외에 Web 및 REST API를 통한 접속 |

**Sqoop1 설치하기**

Sqoop 사이트 : <https://sqoop.apache.org/>

다운로드 경로 : <http://apache.tt.co.kr/sqoop/1.4.7/sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz>

**Sqoop 다운로드 (다운로드 경로는 home)**

$ wget http://apache.tt.co.kr/sqoop/1.4.7/sqoop-1.4.7.bin\_\_hadoop-2.6.0.tar.gz

**압축해제**

$ tar xvzf sqoop-1.4.7.bin\_\_hadoop-2.6.0.tar.gz

**바로가기 생성**

$ ln -s sqoop-1.4.7.bin\_\_hadoop-2.6.0 sqoop

**환경변수 설정 (내용 추가.)**

& sudo gedit ~/.bashrc

|  |
| --- |
| export SQOOP\_HOME=/home/vi/sqoop  export PATH=$PATH:$SQOOP\_HOME/bin |

**환경변수 적용**

& source ~/.bashrc

**Sqoop 환경 설정 (/sqoop/sqoop-1.4.7)**

**파일 이름 변경**

~/sqoop/conf$ cp sqoop-env-template.sh sqoop-env.sh

|  |
| --- |
| #Set path to where bin/hadoop is available  export HADOOP\_COMMON\_HOME=/home/vi(계정이름)/hadoop  #Set path to where hadoop-\*-core.jar is available일  export HADOOP\_MAPRED\_HOME=/home/vi(계정이름)/hadoop |

Sqoop 실행에 필요한 파일.

\* mysql-connector-java-5.1.38-bin.jar 파일을 /home/vi/sqoop/lib 폴더 안에 존재해야 함.

<https://jar-download.com/artifacts/mysql/mysql-connector-java/5.1.38/source-code>

\* 위 사이트를 통해 다운로드 받거나 깃허브 참조.

\* 아래와 같은 오류 발생을 방지하기 위해, commons-lang-2.6.jar 파일 역시 /home/vi/sqoop/lib 폴더 안에 존재해야 함.

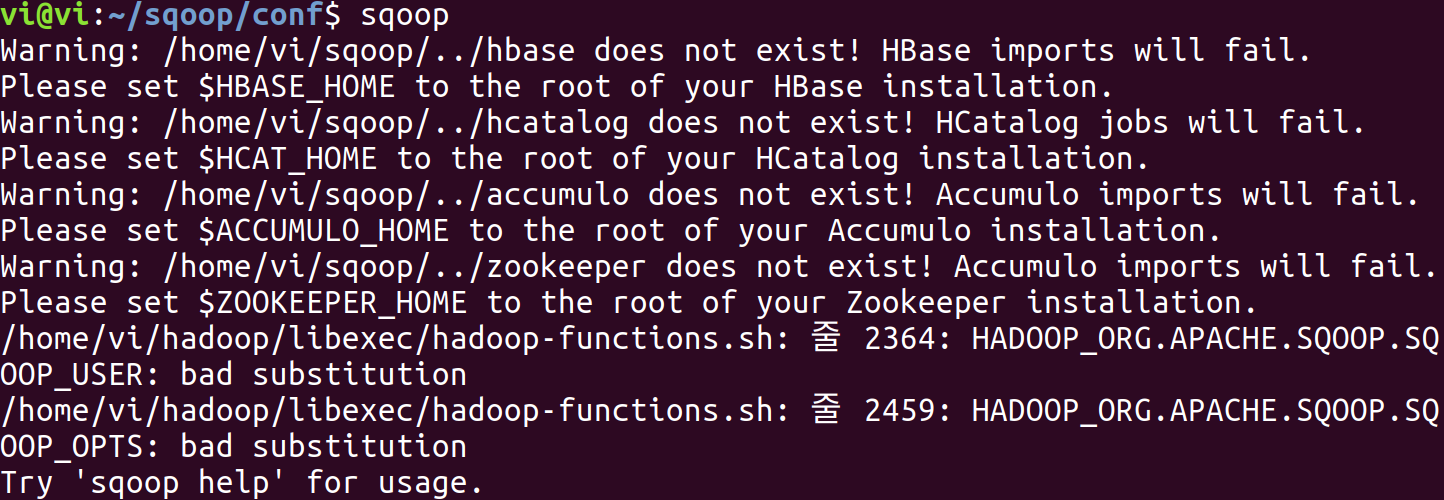
<https://commons.apache.org/proper/commons-lang/download_lang.cgi>

**\*위 사이트를 통해 다운로드 받거나 깃허브 참조.**

|  |
| --- |
| Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/commons/lang/StringUtils |

**스쿱 실행**

|  |
| --- |
| vi@vi:~/sqoop/conf$ sqoop  Warning: /home/vi/sqoop/../hbase does not exist! HBase imports will fail.  Please set $HBASE\_HOME to the root of your HBase installation.  Warning: /home/vi/sqoop/../hcatalog does not exist! HCatalog jobs will fail.  Please set $HCAT\_HOME to the root of your HCatalog installation.  Warning: /home/vi/sqoop/../accumulo does not exist! Accumulo imports will fail.  Please set $ACCUMULO\_HOME to the root of your Accumulo installation.  Warning: /home/vi/sqoop/../zookeeper does not exist! Accumulo imports will fail.  Please set $ZOOKEEPER\_HOME to the root of your Zookeeper installation.  /home/vi/hadoop/libexec/hadoop-functions.sh: 줄 2364: HADOOP\_ORG.APACHE.SQOOP.SQOOP\_USER: bad substitution  /home/vi/hadoop/libexec/hadoop-functions.sh: 줄 2459: HADOOP\_ORG.APACHE.SQOOP.SQOOP\_OPTS: bad substitution  Try 'sqoop help' for usage. # 이렇게 뜬다면 설치 완료. |



경고에 대해서는 각각, hcatalog, Accumulo, Zookeeeper 설정을 안 해주었기 때문에 발생한다.

**Sqoop 사용하기 (MariaDB에서 HDFS로 파일 전송.)**

$ sqoop import --connect jdbc:mysql://localhost/mysql --username root --password 1234 --table test --columns a --target-dir hdfs://localhost:9000/user/yunhyuck/test -m 1

--connect 명령어 줄에는 JDBC 주소를 입력한다. MariaDB에서는 mysql 기반이기 때문에 jdbc:mariadb 입력하는 것을 주의 한다.

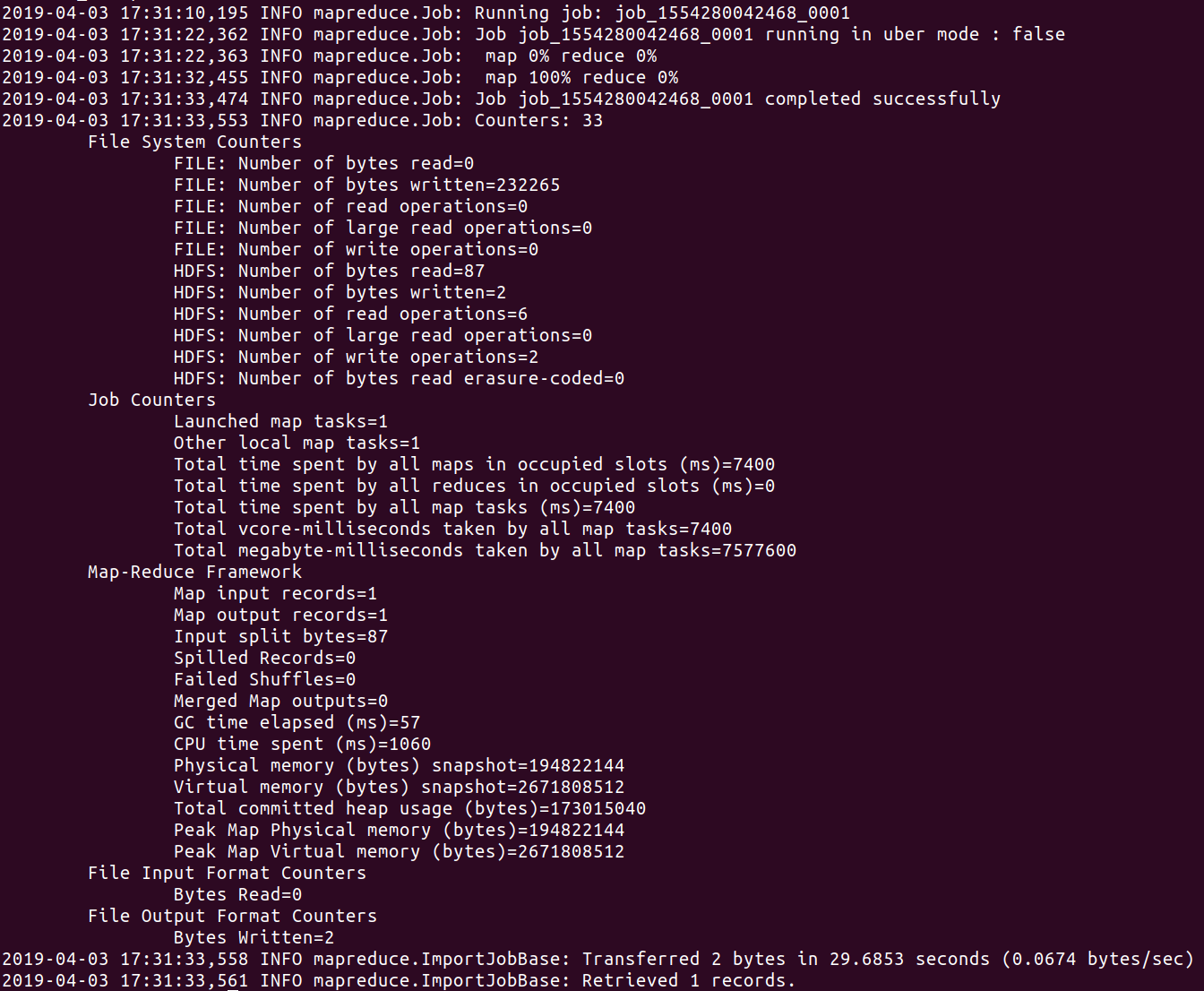
jdbc:mysql://localhost/mysql(테이블 )명

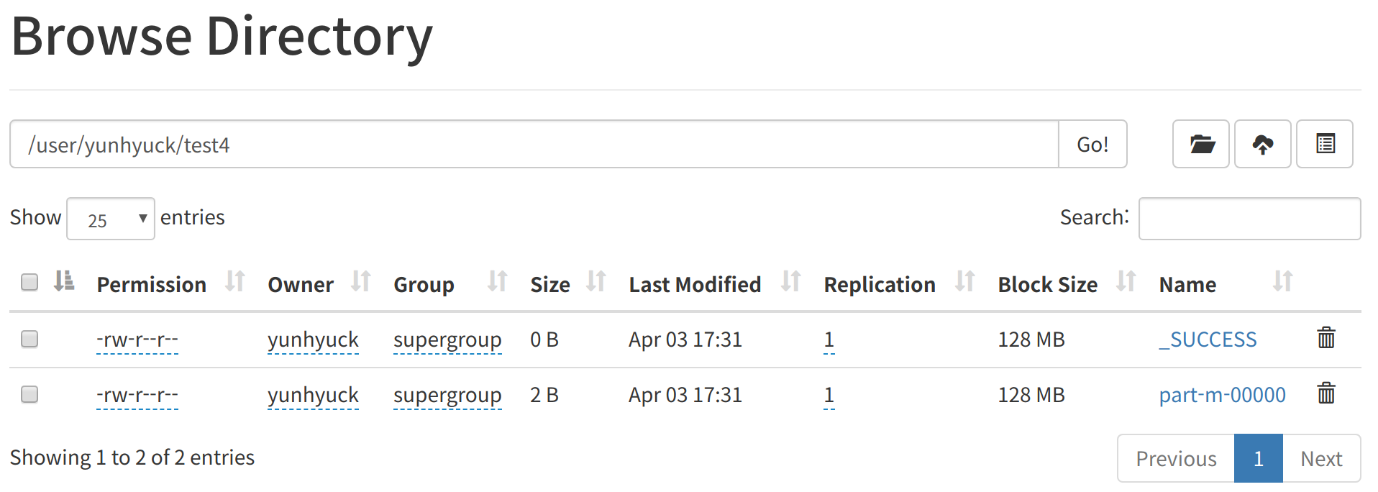
--username root(DB 계정명)

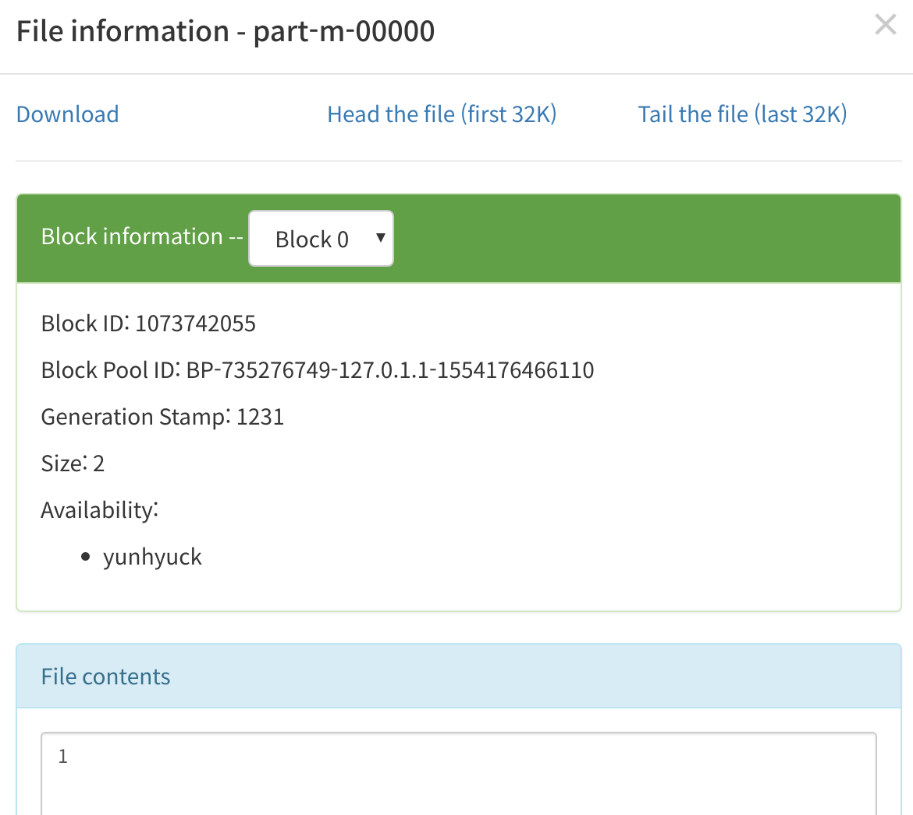
--password xxxx(DB 패스워드)

--table test (DB 테이블 명)

--columns a(DB테이블에서 가져올 칼럼 명)

--target-dir hdfs://localhost:9000/user/yunhyuck/test4 -m 1 (HDFS에 저장하기 위해 HDFS의 주소 와 저장할 위치 지정. 단, 디렉토리는 존재하지 않는 폴더 여야 함.) ****

****



**MariaDB내 테이블 내용**

\* 칼럼 a = text, b = int

test

|  |  |
| --- | --- |
| a | b |
| 이윤혁 | 24 |
| 가나다라마바사아자차카타파하 | 24123 |

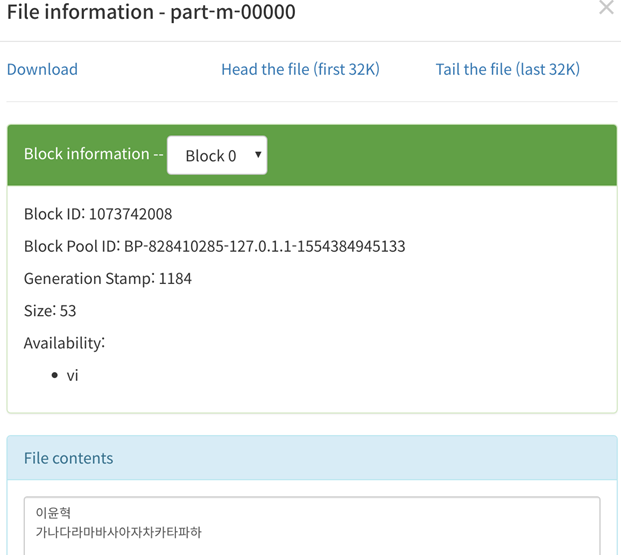
**실험 1. 칼럼을 a,b 둘 다 주었을 경우.**

vi@vi:~$ sqoop import --connect jdbc:mysql://localhost/mysql --username root --password 1234 --table test --columns a,b --target-dir hdfs://localhost:9000/user/yunhyuck/test -m 1



실험 2. 컬럼을 a 만 주었을 경우.

vi@vi:~$ sqoop import --connect jdbc:mysql://localhost/mysql --username root --password 1234 --table test --columns a --target-dir hdfs://localhost:9000/user/yunhyuck/test1 -m 1



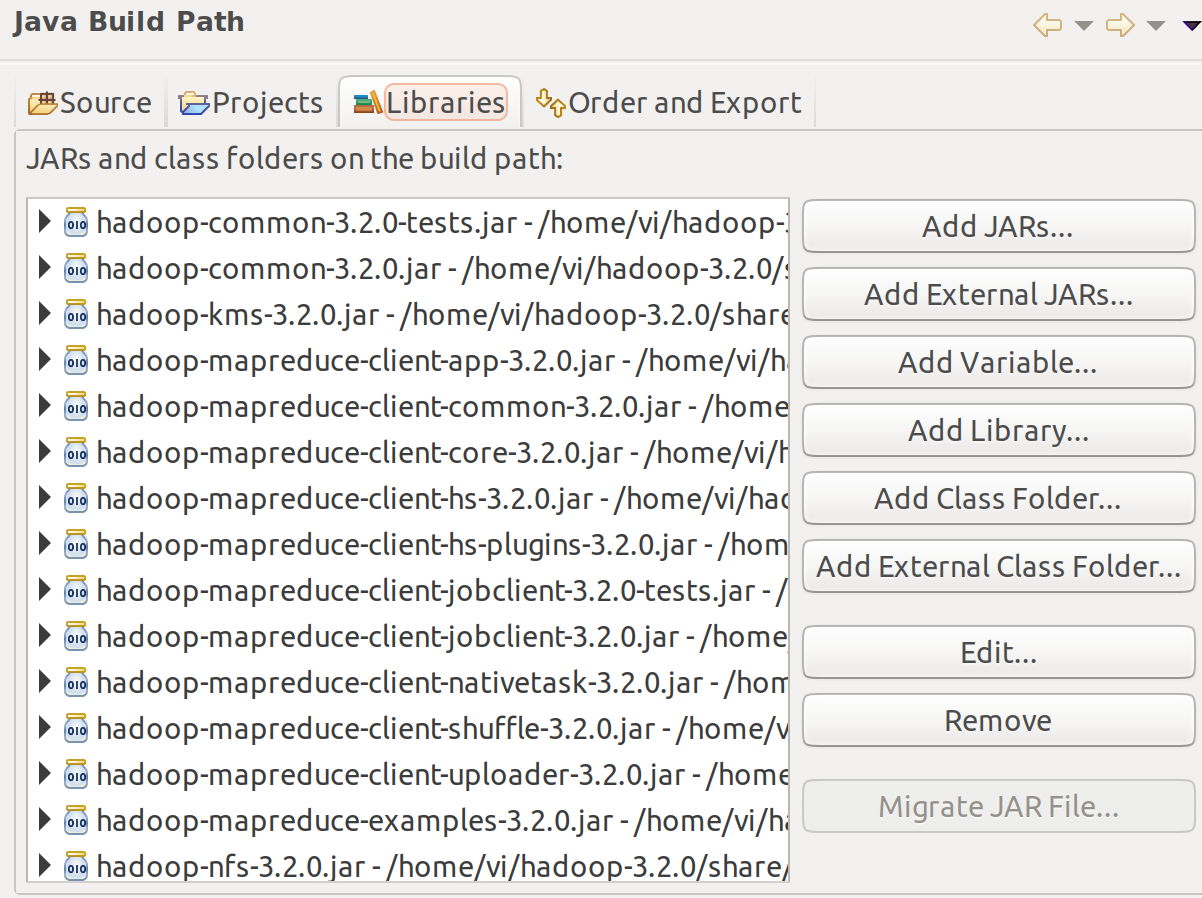
**Jar 파일 생성 부터 하둡 실행까지**

1. Elipse 프로젝트 생성 (프로젝트 명 : KeywordCount)

package는 생성 x.

2. KeyWordCount, KeyWordCountMapper, KeyWordCountReducer 자바 파일을 복사 붙여 넣기.

3. 프로젝트 명에서 마우스 오른쪽 클릭(Build Path –> Configure build Path…)



a. Add External JARs...를 누른다.

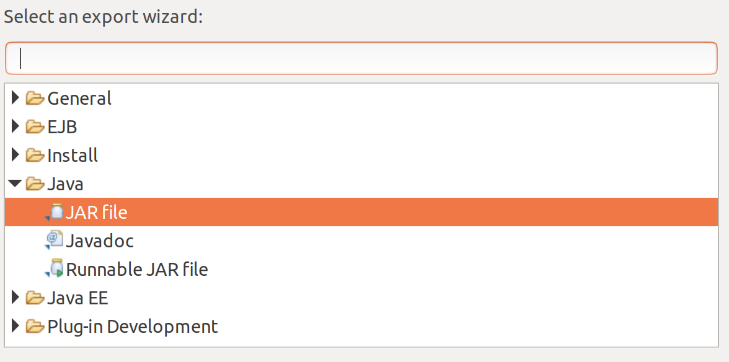
b. hadoop-3.2.0의 Share/hadoop/common 폴더에 있는 jar 파일 추가.

c. hadoop-3.2.0의 Share/hadoop/mapreduce 폴더에 있는 jar 파일 추가.

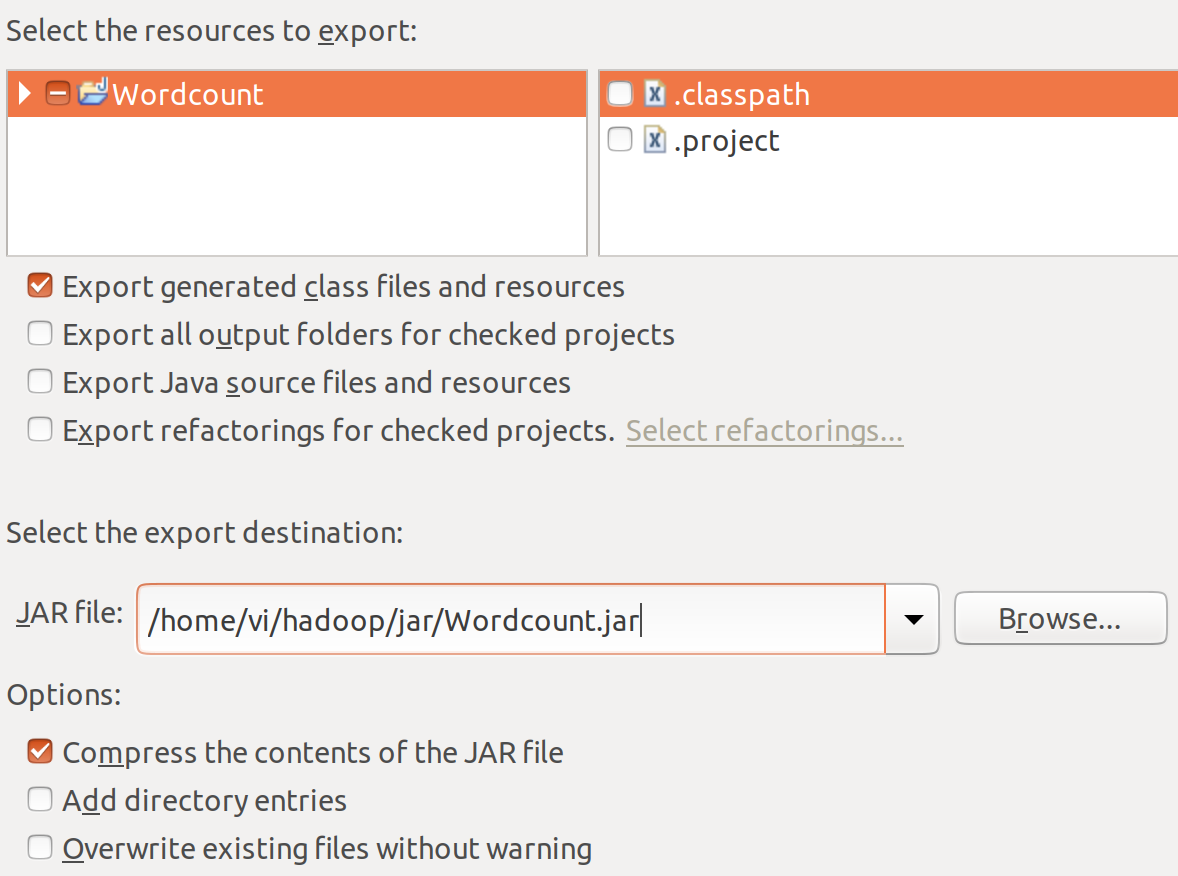
d. Apply and close 클릭.

4. jar 파일 만들기

a. 프로젝트 명에서 마우스 오른쪽 클릭(Export… 클릭)



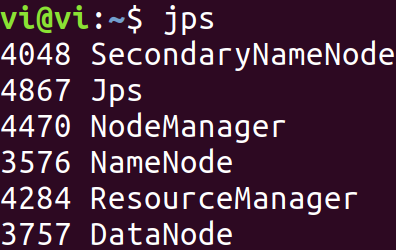
b. 위의 그림과 같이 Java-JAR file 선택 후, NEXT 버튼 클릭



c. .classpath, .project 체크를 풀고, JAR file에서 JAR이 생성될 경로와 파일 명 입력.

d. 확인 후, finish 버튼 클릭.

5. 하둡으로 실행해 보기.

  
\* 하둡은 당연히 켜져 있는 상태.

|  |
| --- |
| **현재 HDFS에 저장되어 있는 내용. (MariaDB에서 테이블 생성 후, 여러 열을 가져 왔을 때)**  **HDFS 파일 경로 명 : /user/yunhyuck/test2/part-m-00000**  **-----------------------------------------**  100,P1,Computer,C413,A  100,P1,Computer,E412,A  200,P2,Electric,C123,B  300,P3,Computer,C312,A  300,P3,Computer,C324,C  300,P3,Computer,C413,A  400,P1,Computer,C312,A  400,P1,Computer,C324,A  400,P1,Computer,C413,B  400,P1,Computer,E412,C |

테스트 데이터가 있다고 가정하고, 실행해 본다. 현재 Map에서 공백을 기준으로 단어를 카운트 한다. 그렇기 때문에 맵 설정에서 replace 명령을 통해 ‘ , ’를 공백으로 바꿔 주고 실행을 한다.

a. yarn 실행

$ yarn jar /home/vi/hadoop/jar/KeyWordcount.jar KeyWordCount /user/yunhyuck/test2/part-m-00000 output

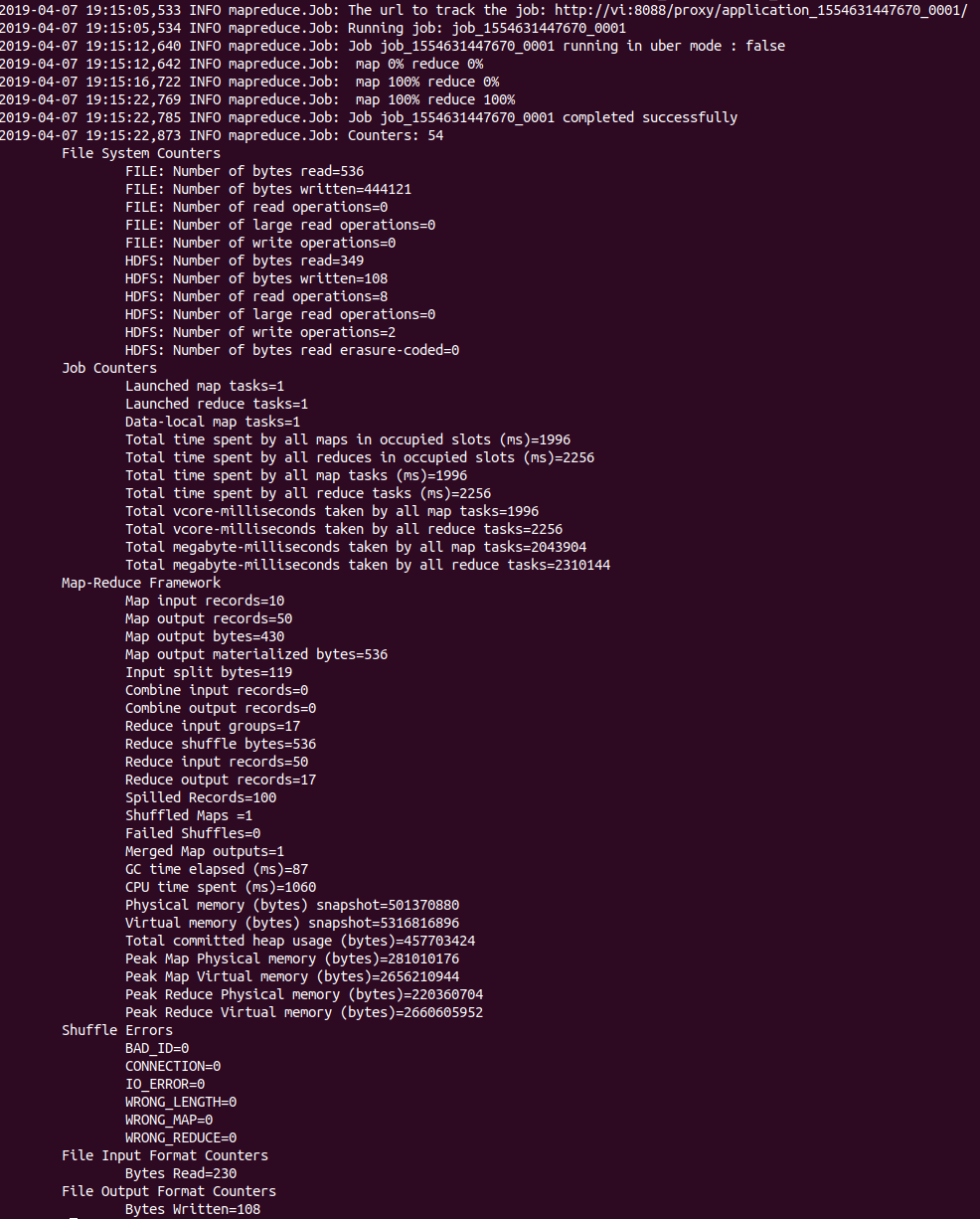
/home/vi/hadoop/jar/KeyWordcount.jar → JAR 파일이 저장된 경로.

WordCount → 드라이버 클래스(KeyWordCount.java에서 잡 객체를 생성할 때, WordCount로 지정. )

/user/yunhyuck/test2/part-m-00000 → 입력 파일 경로

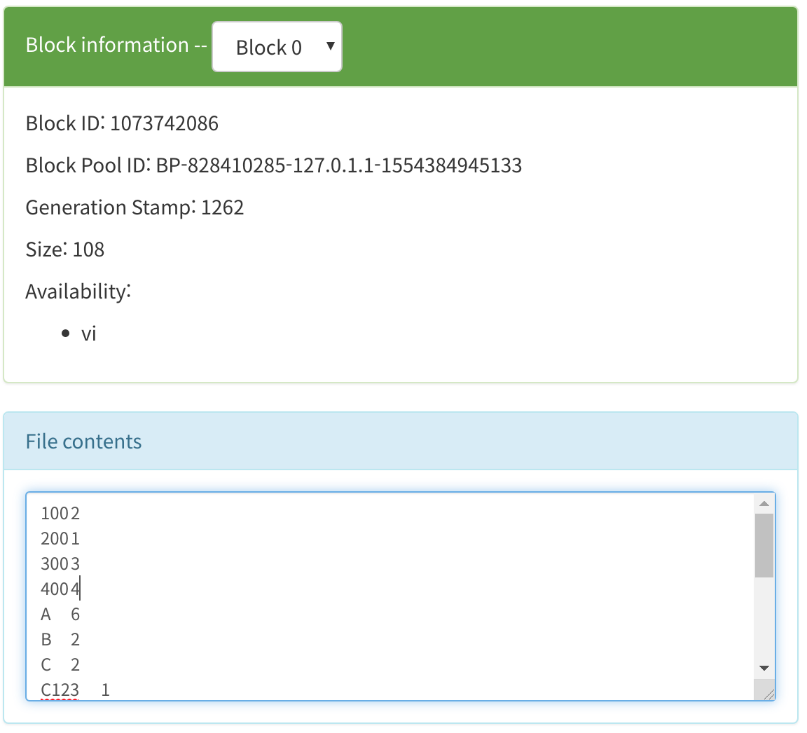
output2 → 출력 파일 경로. 상세 경로를 적어 주지 않으면, /user/vi/output 경로에 출력 파일 생성.

|  |
| --- |
| 2019-04-07 19:15:12,642 INFO mapreduce.Job: map 0% reduce 0%  2019-04-07 19:15:16,722 INFO mapreduce.Job: map 100% reduce 0%  2019-04-07 19:15:22,769 INFO mapreduce.Job: map 100% reduce 100%  2019-04-07 19:15:22,785 INFO mapreduce.Job: Job job\_1554631447670\_0001 completed successfully  2019-04-07 19:15:22,873 INFO mapreduce.Job: Counters: 54  File System Counters  FILE: Number of bytes read=536  FILE: Number of bytes written=444121  FILE: Number of read operations=0  FILE: Number of large read operations=0  FILE: Number of write operations=0  HDFS: Number of bytes read=349  HDFS: Number of bytes written=108  HDFS: Number of read operations=8  HDFS: Number of large read operations=0  HDFS: Number of write operations=2  HDFS: Number of bytes read erasure-coded=0  Job Counters  Launched map tasks=1  Launched reduce tasks=1  Data-local map tasks=1  Total time spent by all maps in occupied slots (ms)=1996  Total time spent by all reduces in occupied slots (ms)=2256  Total time spent by all map tasks (ms)=1996  Total time spent by all reduce tasks (ms)=2256  Total vcore-milliseconds taken by all map tasks=1996  Total vcore-milliseconds taken by all reduce tasks=2256  Total megabyte-milliseconds taken by all map tasks=2043904  Total megabyte-milliseconds taken by all reduce tasks=2310144  Map-Reduce Framework  Map input records=10  Map output records=50  Map output bytes=430  Map output materialized bytes=536  Input split bytes=119  Combine input records=0  Combine output records=0  Reduce input groups=17  Reduce shuffle bytes=536  Reduce input records=50  Reduce output records=17  Spilled Records=100  Shuffled Maps =1  Failed Shuffles=0  Merged Map outputs=1  GC time elapsed (ms)=87  CPU time spent (ms)=1060  Physical memory (bytes) snapshot=501370880  Virtual memory (bytes) snapshot=5316816896  Total committed heap usage (bytes)=457703424  Peak Map Physical memory (bytes)=281010176  Peak Map Virtual memory (bytes)=2656210944  Peak Reduce Physical memory (bytes)=220360704  Peak Reduce Virtual memory (bytes)=2660605952  Shuffle Errors  BAD\_ID=0  CONNECTION=0  IO\_ERROR=0  WRONG\_LENGTH=0  WRONG\_MAP=0  WRONG\_REDUCE=0  File Input Format Counters  Bytes Read=230  File Output Format Counters  Bytes Written=108 |



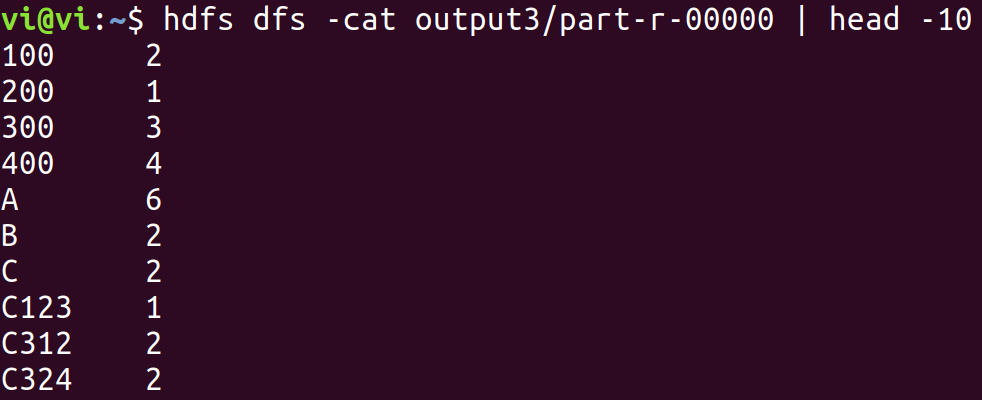
<http://localhost:9870>

접속 해서 출력 파일 확인.



**MapReduce로 처리 되어 HDFS에 저장된 파일을 MariaDB로 저장 하기**

**MapReduce처리된 파일 확인**

vi@vi:~$ hdfs dfs -cat output3/part-r-00000 | head -10

|  |
| --- |
| 100 2  200 1  300 3  400 4  A 6  B 2  C 2  C123 1  C312 2  C324 2 |

**Sqoop을 통한 MariaDB 파일 저장**

vi@vi:~$ sqoop export --connect jdbc:mysql://localhost/mysql --username root --password 1234 --table test --export-dir hdfs://localhost:9000/user/vi/output3/part-r-00000 --columns a,b --input-fields-terminated-by "\t"

--connect jdbc:mysql://localhost/mysql -> MariaDB의 DB 명

--username root --password 1234 -> MariaDB의 ID, Password

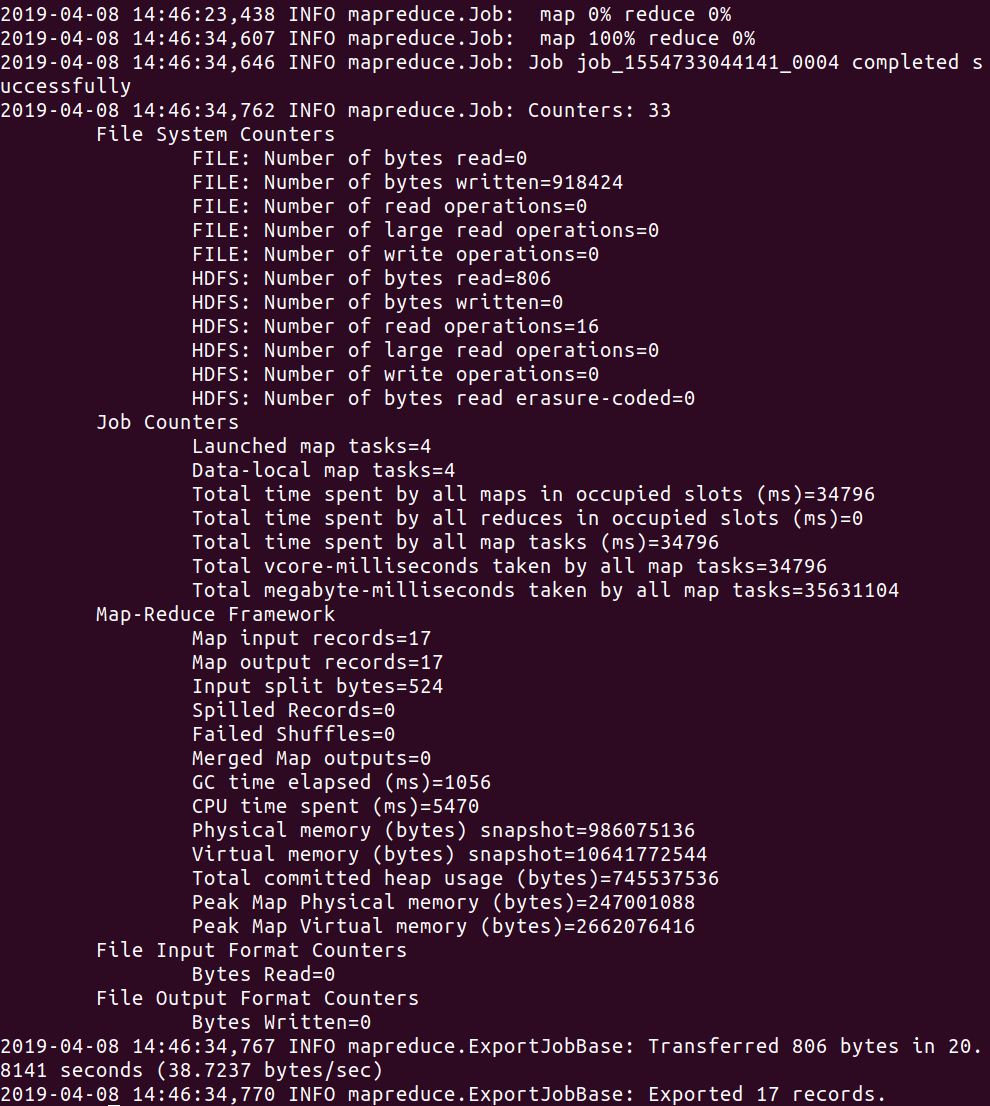
--table test -> MariaDB의 테이블 명(존재 하고 있어야 함.)

--export-dir hdfs://localhost:9000/user/vi/output3/part-r-00000 -> input 파일 경로

--columns a,b -> MariaDB에 저장될 컬럼

--input-fields-terminated-by "\t" -> input 데이터가 tab으로 구분되어 있으므로 구분 지어 주는 역할.

|  |
| --- |
| 2019-04-08 14:46:17,242 INFO mapreduce.Job: Running job: job\_1554733044141\_0004  2019-04-08 14:46:23,436 INFO mapreduce.Job: Job job\_1554733044141\_0004 running in uber mode : false  2019-04-08 14:46:23,438 INFO mapreduce.Job: map 0% reduce 0%  2019-04-08 14:46:34,607 INFO mapreduce.Job: map 100% reduce 0%  2019-04-08 14:46:34,646 INFO mapreduce.Job: Job job\_1554733044141\_0004 completed successfully  2019-04-08 14:46:34,762 INFO mapreduce.Job: Counters: 33  File System Counters  FILE: Number of bytes read=0  FILE: Number of bytes written=918424  FILE: Number of read operations=0  FILE: Number of large read operations=0  FILE: Number of write operations=0  HDFS: Number of bytes read=806  HDFS: Number of bytes written=0  HDFS: Number of read operations=16  HDFS: Number of large read operations=0  HDFS: Number of write operations=0  HDFS: Number of bytes read erasure-coded=0  Job Counters  Launched map tasks=4  Data-local map tasks=4  Total time spent by all maps in occupied slots (ms)=34796  Total time spent by all reduces in occupied slots (ms)=0  Total time spent by all map tasks (ms)=34796  Total vcore-milliseconds taken by all map tasks=34796  Total megabyte-milliseconds taken by all map tasks=35631104  Map-Reduce Framework  Map input records=17  Map output records=17  Input split bytes=524  Spilled Records=0  Failed Shuffles=0  Merged Map outputs=0  GC time elapsed (ms)=1056  CPU time spent (ms)=5470  Physical memory (bytes) snapshot=986075136  Virtual memory (bytes) snapshot=10641772544  Total committed heap usage (bytes)=745537536  Peak Map Physical memory (bytes)=247001088  Peak Map Virtual memory (bytes)=2662076416  File Input Format Counters  Bytes Read=0  File Output Format Counters  Bytes Written=0  2019-04-08 14:46:34,767 INFO mapreduce.ExportJobBase: Transferred 806 bytes in 20.8141 seconds (38.7237 bytes/sec)  2019-04-08 14:46:34,770 INFO mapreduce.ExportJobBase: Exported 17 records. |



**MariaDB 를 통한 데이터 확인.**

|  |
| --- |
| MariaDB [mysql]> create table test(a text, b int);  Query OK, 0 rows affected (0.08 sec)  MariaDB [mysql]> select \*from test;  Empty set (0.01 sec)  ---------------------------------------------------------------------------------------------------------------------------------  MariaDB [mysql]> select \*from test;  +----------+------+  | a | b |  +----------+------+  | B | 2 |  | C | 2 |  | C123 | 1 |  | C312 | 2 |  | C324 | 2 |  | Electric | 1 |  | P1 | 6 |  | P2 | 1 |  | P3 | 3 |  | C413 | 3 |  | Computer | 9 |  | E412 | 2 |  | 100 | 2 |  | 200 | 1 |  | 300 | 3 |  | 400 | 4 |  | A | 6 |  +----------+------+  17 rows in set (0.00 sec) |

