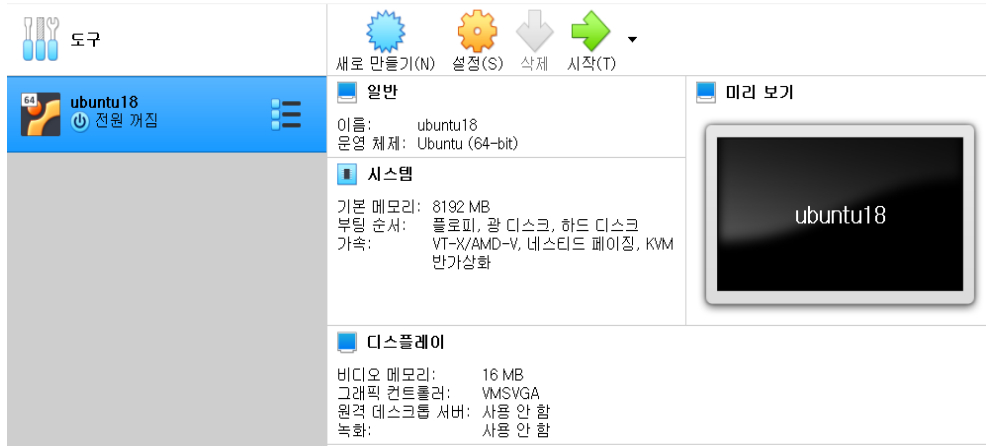
Ubuntu, Hadoop_Setting, WordCount & Hive 실행

최선종

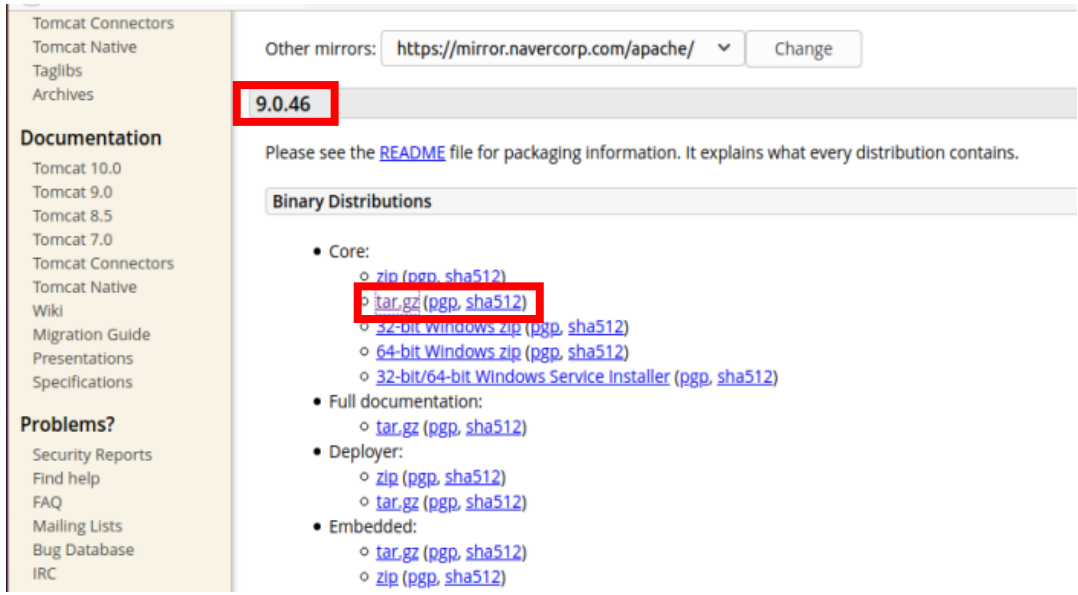
Ubuntu Setting – xml 및 사용자 계정 설정

Ubuntu설치

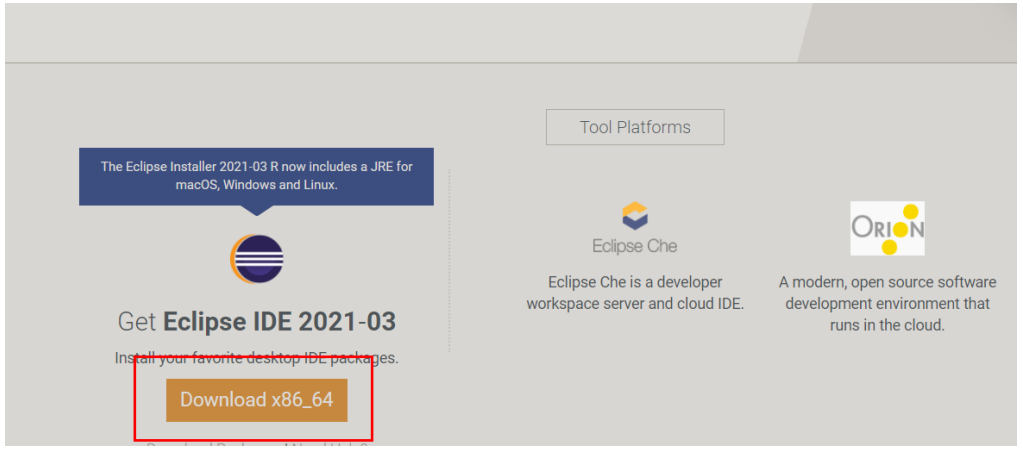


Oracle VirtualBox 다운받고 사용자 생성
메모리 공간 및 하드웨어는 시스템 환경에 따라
유동적으로 설정

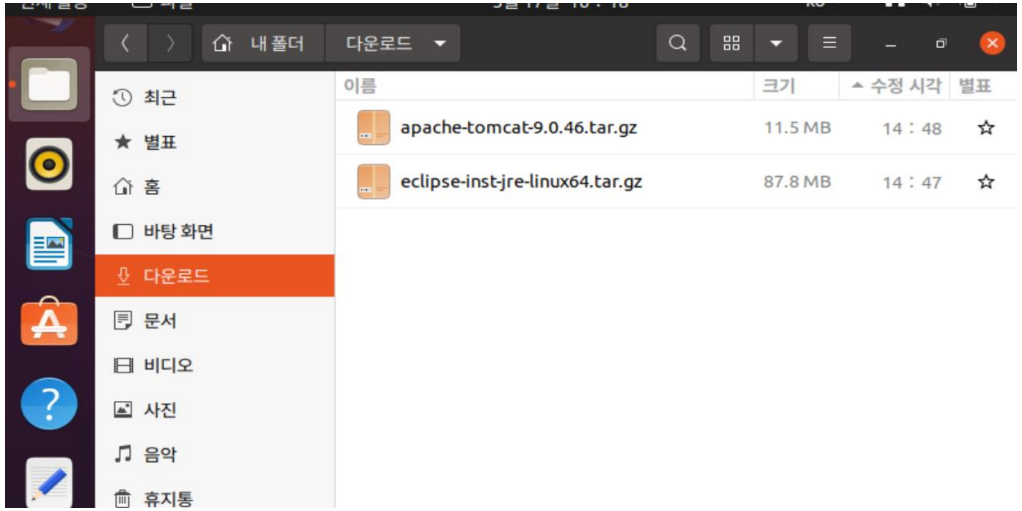
Apache-tomcat 설치



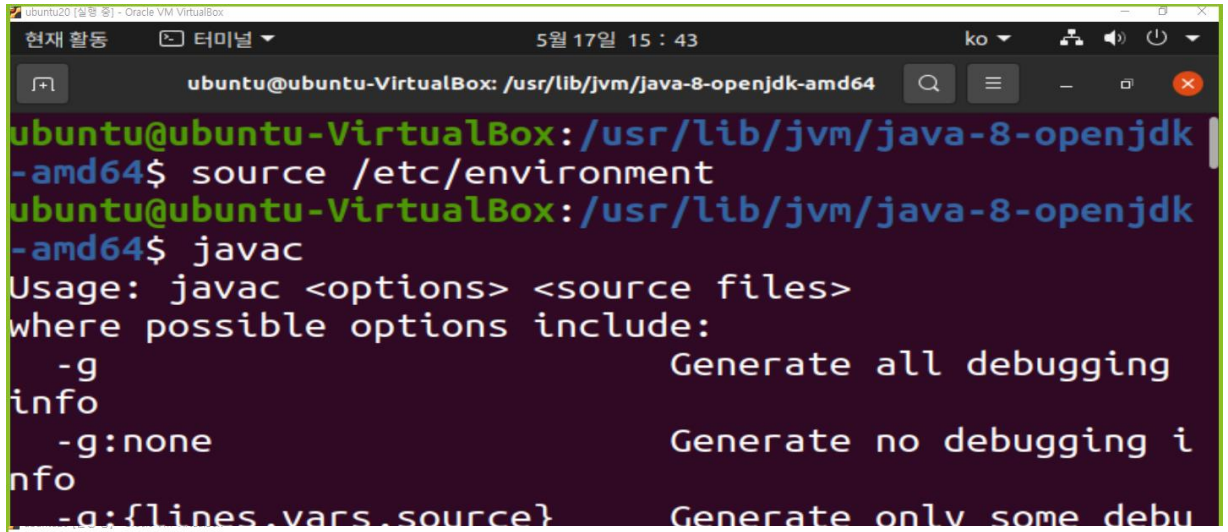
이클립스 설치

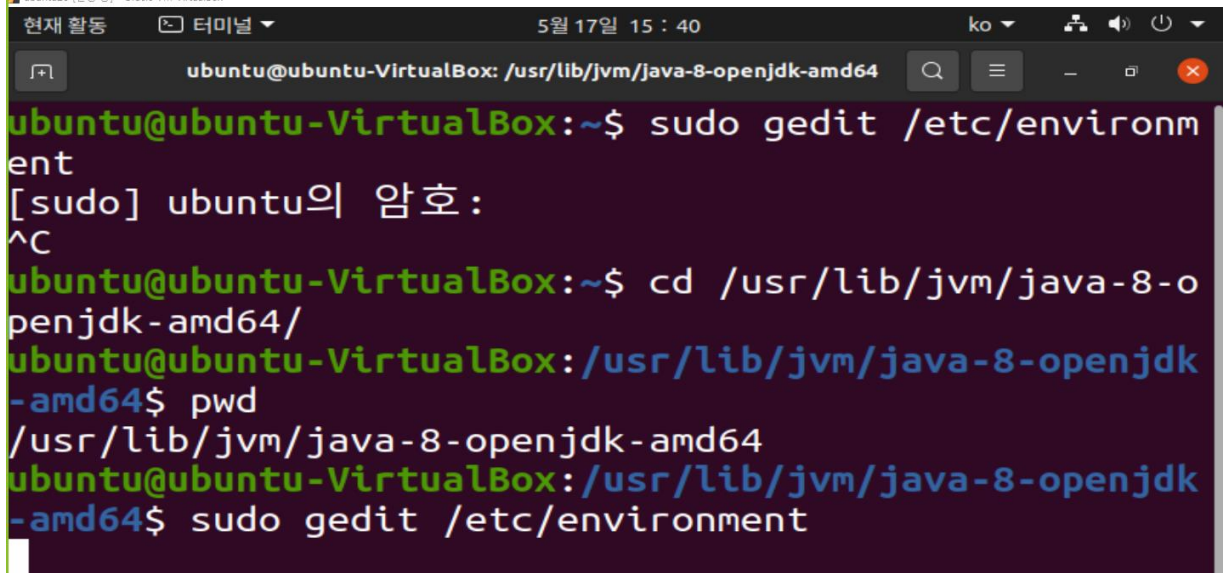


다운로드 완료

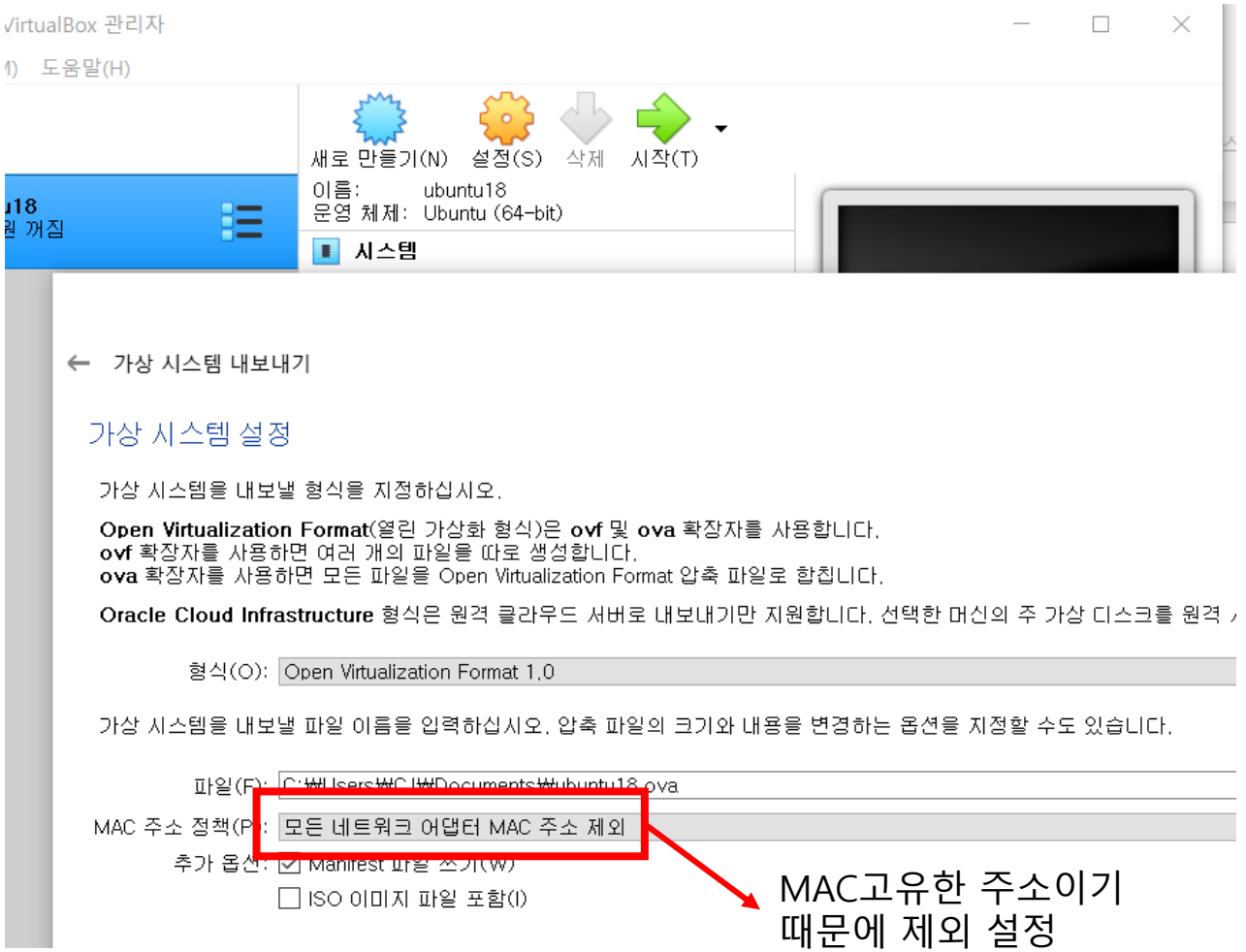


Ubuntu Setting– xml 및 사용자 계정 설정





관리자 권한으로 폴더 접근하여
Java 환경변수 지정을 위해 path 설정



가상 시스템 내보내기 실행하여
이미지 파일 생성(해당 이미 파일로 어디서든 접속 가능)

Hadoop Setting – 파일 다운/압축 풀기/파일 이동/bashrc 설정

Hadoop 다운로드

Hadoop-2.9.1 폴더 → /usr/local/Hadoop 폴더로 이동

```
ubuntu@ubuntu-VirtualBox:~$ cd download
bash: cd: download: 그런 파일이나 디렉터리가 없습니다
ubuntu@ubuntu-VirtualBox:~$ ls
eclipse            hadoop-2.9.1.tar.gz  다운로드  바탕화면  사진  템플릿
eclipse-workspace  공개                 문서      비디오    음악
ubuntu@ubuntu-VirtualBox:~$ cd hadoop-2.9.1/
bash: cd: hadoop-2.9.1/: 그런 파일이나 디렉터리가 없습니다
ubuntu@ubuntu-VirtualBox:~$ cd 다운로드/
ubuntu@ubuntu-VirtualBox:~/다운로드$ cd hadoop-2.9.1/
ubuntu@ubuntu-VirtualBox:~/다운로드/hadoop-2.9.1$ pwd
/home/ubuntu/다운로드/hadoop-2.9.1
ubuntu@ubuntu-VirtualBox:~/다운로드/hadoop-2.9.1$ sudo mkdir /usr/local/hadoop
[sudo] ubuntu의 암호:
mkdir: `/usr/local/hadoop' 디렉토리를 만들 수 없습니다: 파일이 있습니다
ubuntu@ubuntu-VirtualBox:~/다운로드/hadoop-2.9.1$ sudo mkdir /usr/local/hadoop
mkdir: `/usr/local/hadoop' 디렉토리를 만들 수 없습니다: 파일이 있습니다
ubuntu@ubuntu-VirtualBox:~/다운로드/hadoop-2.9.1$ sudo mv * /usr/local/hadoop/
ubuntu@ubuntu-VirtualBox:~/다운로드/hadoop-2.9.1$ ls
ubuntu@ubuntu-VirtualBox:~/다운로드/hadoop-2.9.1$ cd ..
ubuntu@ubuntu-VirtualBox:~/다운로드$ cd hadoop-2.9.1/
ubuntu@ubuntu-VirtualBox:~/다운로드/hadoop-2.9.1$ sudo mv * /usr/local/hadoop
```

관리자 권한으로 접속하여 Bashrc 파일 설정

```
ubuntu@ubuntu-VirtualBox:~$ sudo gedit .bashrc

(gedit:34189): Tepl-WARNING **: 12:51:30.595: GVfs metadata is no
to TeplMetadataManager. Either GVfs is not correctly installed or
 supported on this platform. In the latter case, you should confi
le-gvfs-metadata.
ubuntu@ubuntu-VirtualBox:~$ source .bashrc
```

```
5 ############## HADOOP ########################
6 export HADOOP_HOME=/usr/local/hadoop
7 export PATH=$PATH:$HADOOP_HOME/bin
8 export PATH=$PATH:$HADOOP_HOME/sbin
9 export HADOOP_MAPRED_HOME=$HADOOP_HOME
10 export HADOOP_COMMON_HOME=$HADOOP_HOME
11 export HADOOP_HDFS_HOME=$HADOOP_HOME
12 export YARN_HOME=$HADOOP_HOME
13 export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
14 export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
```

Hadoop Setting – xml 및 사용자 계정 설정

XML 설정

core-site.xml
/usr/local/hadoop/etc/hadoop

```
14   limitations under the License. See accompanying L
15 -->
16
17 <!-- Put site-specific property overrides in this f
18
19 <configuration>
20
21 <property>
22
23     <name>hadoop.tmp.dir</name>
24
25     <value>/usr/local/hadoop/tmp</value>
26
27 </property>
28
29 <property>
30
31     <name>fs.default.name</name>
32
33     <value>hdfs://localhost:62350</value>
34
```

hdfs-site.xml
/usr/local/hadoop/etc/hadoop

```
35 </property>
36
37 <property>
38
39     <name>dfs.datanode.data.dir</name>
40
41     <value>file:/usr/local/hadoop/hdfs/datanode</value>
42
43 </property>
44
45 <property>
46
47     <name>dfs.http.address</name>
48
49     <value>localhost:50070</value>
50
51 </property>
52
53 <property>
54
55     <name>dfs.secondary.http.address</name>
56
57     <value>localhost:50090</value>
58
```

mapred-site.xml
/usr/local/hadoop/etc/hadoop

```
6   You may obtain a copy of the License at
7
8     http://www.apache.org/licenses/LICENSE-2.0
9
10  Unless required by applicable law or agreed to in writing, so
11  distributed under the License is distributed on an "AS IS" BA
12  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
13  See the License for the specific language governing permissio
14  limitations under the License. See accompanying LICENSE file.
15 -->
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20
21 <property>
22
23     <name>mapred.job.tracker</name>
24
25     <value>localhost:62351</value>
26
27 </property>
28
29 </configuration>
```

hadoop-env.sh
/usr/local/hadoop/etc/hadoop

```
7 # with the License.  You may obtain a copy of the License at
8 #
9 #     http://www.apache.org/licenses/LICENSE-2.0
10 #
11 # Unless required by applicable law or agreed to in writing, software
12 # distributed under the License is distributed on an "AS IS" BASIS,
13 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 # See the License for the specific language governing permissions and
15 # limitations under the License.
16
17 # Set Hadoop-specific environment variables here.
18
19 # The only required environment variable is JAVA_HOME.  All others are
20 # optional.  When running a distributed configuration it is best to
21 # set JAVA_HOME in this file, so that it is correctly defined on
22 # remote nodes.
23
24 # The java implementation to use.
25 export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
26
27 # The jsvc implementation to use. Jsvc is required to run secure datanodes
28 # that bind to privileged ports to provide authentication of data transfer
29 # protocol.  Jsvc is not required if SASL is configured for authentication of
30 # data transfer protocol using non-privileged ports.
```

사용자 및 slave 설정 확인

```
ubuntu@ubuntu-VirtualBox:~$ sudo addgroup hadoop
[sudo] ubuntu의 암호:
그룹 `hadoop' (GID 1001) 추가 ...
완료.
ubuntu@ubuntu-VirtualBox:~$ sudo adduser ubuntu hadoop
'ubuntu' 사용자를 'hadoop' 그룹에 추가 중...
사용자 ubuntu을(를) hadoop 그룹에 등록 중
완료.
ubuntu@ubuntu-VirtualBox:~$
```

```
ubuntu@ubuntu-VirtualBox:~$ cd /usr/local/hadoop/etc/hadoop/
ubuntu@ubuntu-VirtualBox:/usr/local/hadoop/etc/hadoop$ sudo gedit slaves
[sudo] ubuntu의 암호:

(gedit:13215): Tepl-WARNING **: 15:40:11.020: GVfs metadata is not suppor
to TeplMetadataManager. Either GVfs is not correctly installed or GVfs me
 supported on this platform. In the latter case, you should configure Tep
le-gvfs-metadata.
```

slaves
/usr/local/hadoop/etc/hadoop

```
1 localhost
```

Hadoop Setting – xml 및 사용자 계정 설정

폴더 생성 및 권한설정

```
ubuntu@ubuntu-VirtualBox:/usr/local/hadoop$ sudo mkdir hdfs
ubuntu@ubuntu-VirtualBox:/usr/local/hadoop$ sudo mkdir tmp
ubuntu@ubuntu-VirtualBox:/usr/local/hadoop$ cd hdfs/
ubuntu@ubuntu-VirtualBox:/usr/local/hadoop/hdfs$ mkdir datanode namenode
```

```
ubuntu@ubuntu-VirtualBox:~$ sudo chown -R ubuntu:hadoop /usr/local/hadoop/
```

```
ubuntu@ubuntu-VirtualBox:/usr/local/hadoop/hdfs$ sudo chown -R ubuntu:hadoop /usr/local
/hadoop/tmp
ubuntu@ubuntu-VirtualBox:/usr/local/hadoop/hdfs$ sudo chown -R ubuntu:hadoop /usr/local
/hadoop/hdfs/namenode
ubuntu@ubuntu-VirtualBox:/usr/local/hadoop/hdfs$ sudo chown -R ubuntu:hadoop /usr/local
/hadoop/hdfs/datanode
```

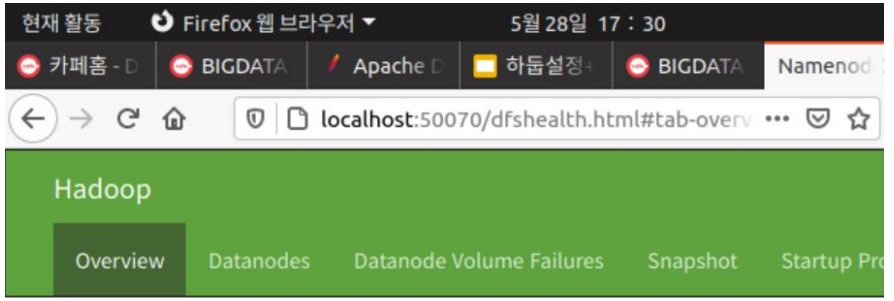
```
ubuntu@ubuntu-VirtualBox:~$ hadoop namenode -format
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
```

start-all.sh 실행

```
ubuntu@ubuntu-VirtualBox:~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
21/05/28 15:26:43 WARN util.NativeCodeLoader: Unable to load native-hadoop
your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
```

jps 및 홈페이지 확인

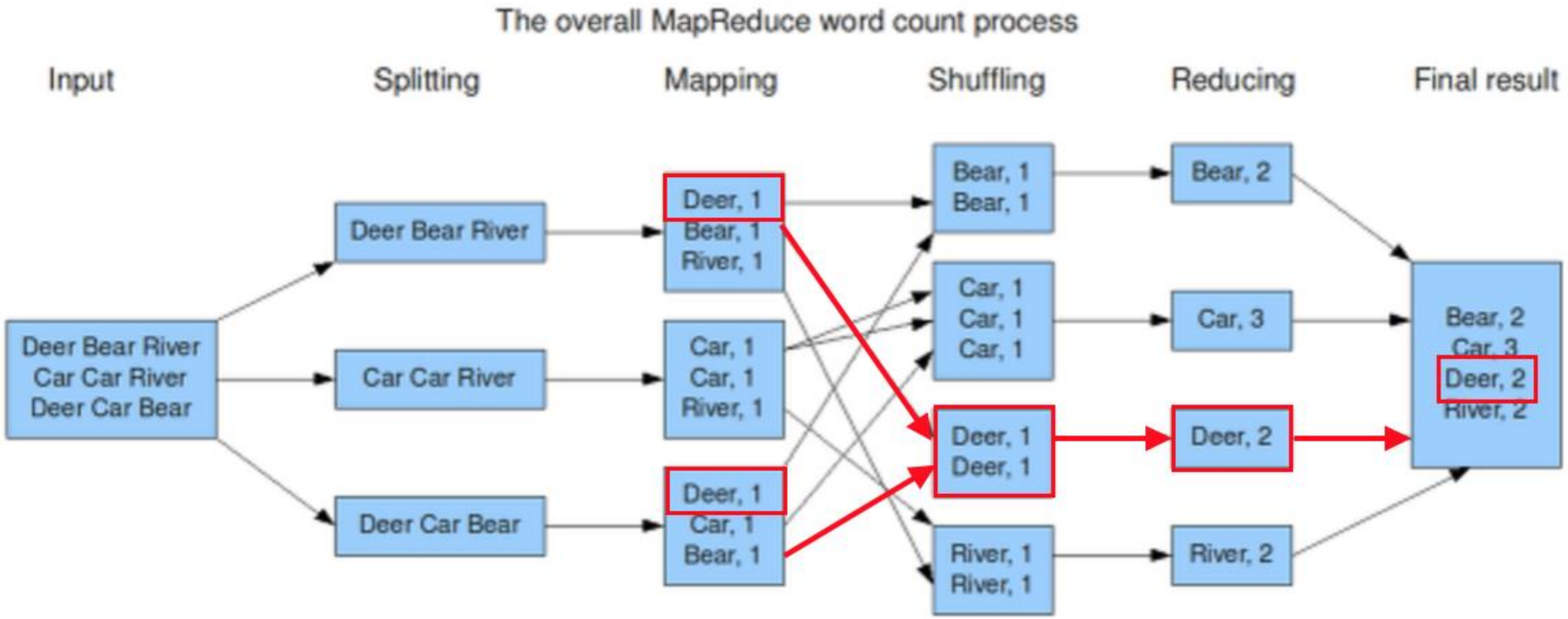
```
ubuntu@ubuntu-VirtualBox:~$ jps
4050 NodeManager
3254 NameNode
3735 ResourceManager
4172 Jps
3613 SecondaryNameNode
3423 DataNode
```



Overview 'localhost:62350' (active)

Started:	Fri May 28 17:28:31 +0900 2021
Version:	2.9.1, re30710aea4e6e55e69372929106cf119af06fd0e
Compiled:	Mon Apr 16 18:33:00 +0900 2018 by root from branch-2.9.1
Cluster ID:	CID-eb73a0db-79bc-499c-a87e-cd3e02fc1e21
Block Pool ID:	BP-175521109-127.0.1.1-1622183131332

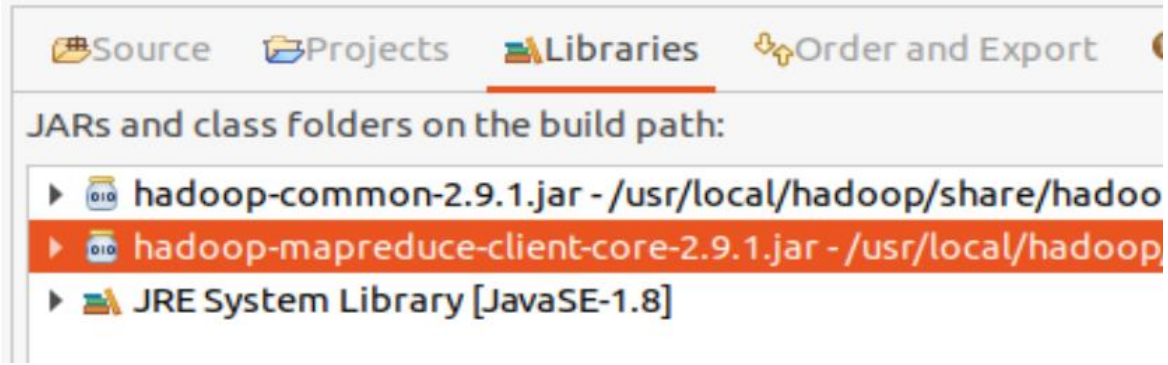
Hadoop wordcount – 개념



단어 input → Map(key, value) 값으로 분류 → 단어 개수에 따라 분류
→ Reduce 단어 개수만큼 더해 줌 → 단어, 빈도수 최종 결과로 주어짐

Hadoop wordcount – 라이브러리 설정

Build Path 라이브러리 설정



WordCount Class

```
public class WordCount2 extends Configured implements Tool{

    //WordCount 분석할파일경로/파일명   분석할파일저장경로
    public static void main(String[] args) throws Exception {
        int exitCode = ToolRunner.run(new WordCount2(), args);
        System.exit(exitCode);
    }
    @Override
    public int run(String[] args) throws Exception {
        //WordCount 하기 위한 객체 생성
        JobConf conf = new JobConf();
        conf.setJobName("=========wordCount=========");
        conf.setOutputKeyClass(Text.class);  //감자
        conf.setOutputValueClass(IntWritable.class);  //5
        conf.setMapperClass(Map2.class);
        conf.setReducerClass(Reduce2.class);
        conf.setInputFormat(TextInputFormat.class);
        conf.setOutputFormat(TextOutputFormat.class);
        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
        JobClient.runJob(conf);
```

생성된 Map, Reduce 클래스 사용하여 Job 실행

Map Class

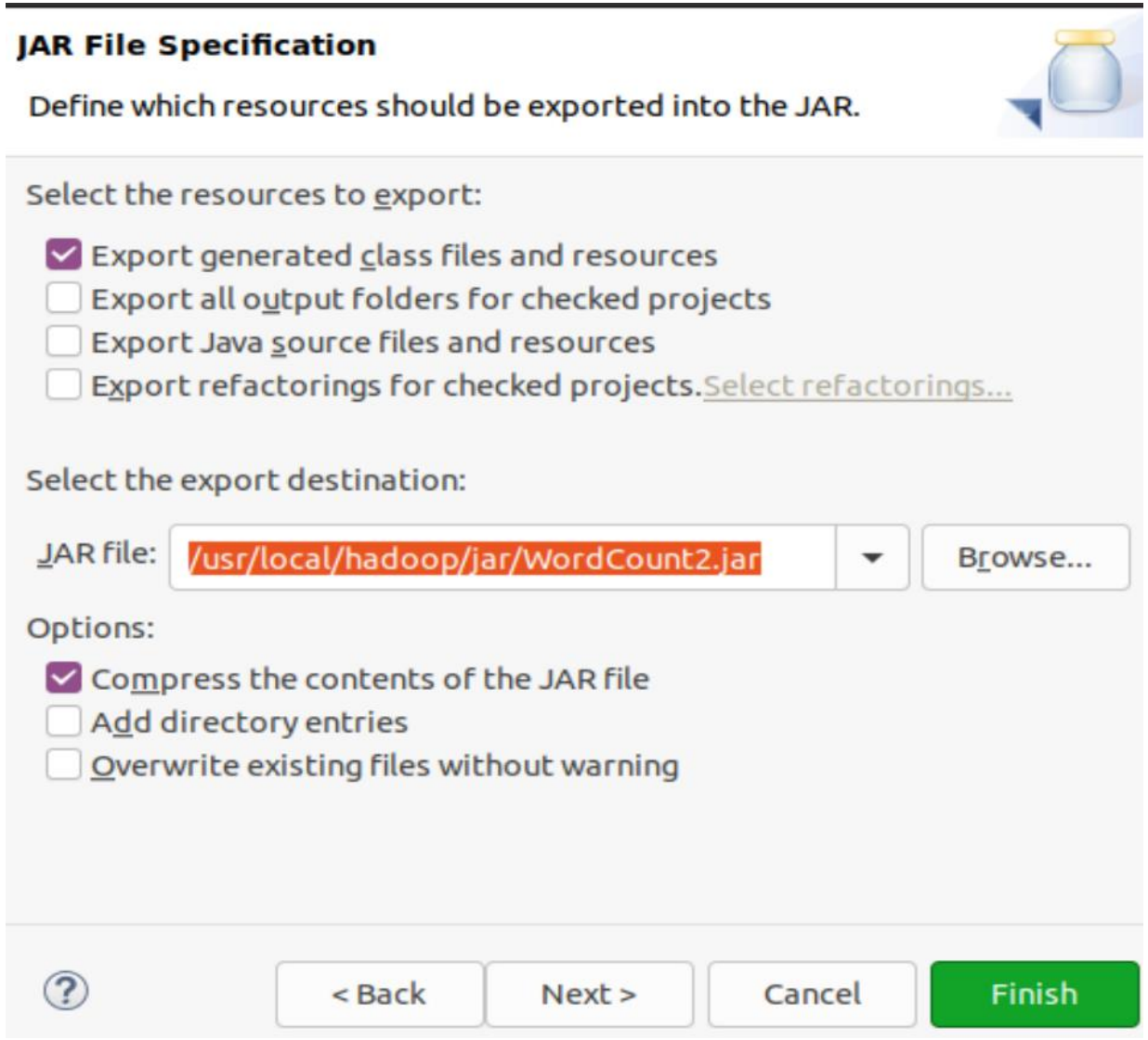
```
package com.mega.wc;

import java.io.IOException;

public class Map2 extends MapReduceBase implements Mapper<LongWritable, Text

    @Override
    public void map(LongWritable blockId, Text value, OutputCollector<Text,

        // block 단위로 입력값이 들어온다. => 100: (value) 감자 고구마 감자 고구마 양파 파
        StringTokenizer st = new StringTokenizer(value.toString().toLowerCase

        // {"감자", "고구마", "감자", "고구마", "양파", "파"}
        while (st.hasMoreTokens()) {

            // 인덱스를 1씩 증가하면서 해당 인덱스 값이 있는지를 체크, true/false
            output.collect(new Text(st.nextToken()), new IntWritable(1));
        }
```

Reduce Class

```
1  package com.mega.wc;
2
3  import java.io.IOException;
1
2  public class Reduce2 extends MapReduceBase implements Reducer<Text, IntWrita
3
4      @Override
5      public void reduce(Text key, Iterator<IntWritable> values, OutputCollecto
6              throws IOException {
7          // 감자 : 1 1 1 1 1 1
8          // 고구마 : 1 1 1
9          int cnt = 0;
0          while (values.hasNext()) {
1              cnt += values.next().get();
2          }
3          output.collect(key, new IntWritable(cnt));
4      }
```

Hadoop wordcount – Export 및 실행

프로젝트 파일 jar 파일로 export



Wordcount 실행과정

```
  104  start-all.sh
  105  jps
  106  clear
  107  hadoop fs -mkdir -P /wordcount/input1
  108  hadoop fs -mkdir -p /wordcount/input1
  109  hadoop fs -copyFromLocal /usr/local/hadoop/README.txt /wordcount/input1
  110  hadoop fs -ls
  111  hadoop fs -ls /
  112  hadoop fs -ls /wordcount/input1
  113  hadoop jar jar/WordCount.jar com.mega.wc.WordCount /wordcount/input1 /wordc
ount/output1
  114  hadoop fs -ls /wordcount/output1
  115  hadoop fs -cat /wordcount/output1/_SUCCESS
  116  hadoop fs -cat /wordcount/output1/part-00000
  117  stop-all.sh
  118  hadoop namenode -format
```

Terminal에서 wordcount 실행하기 위한 전반적인 흐름

Start-all.sh 실행 → jps 확인(namenode, datanode 등 실행하기 위한 준비들이 되었는지 확인) → count할 파일을 넣을 input 폴더 생성 → Count할 파일(README) input 폴더로 이동 → WordCount.jar 파일 실행, ouput 폴더 생성하여 실행결과 넣기 → 실행결과 확인 및 종료

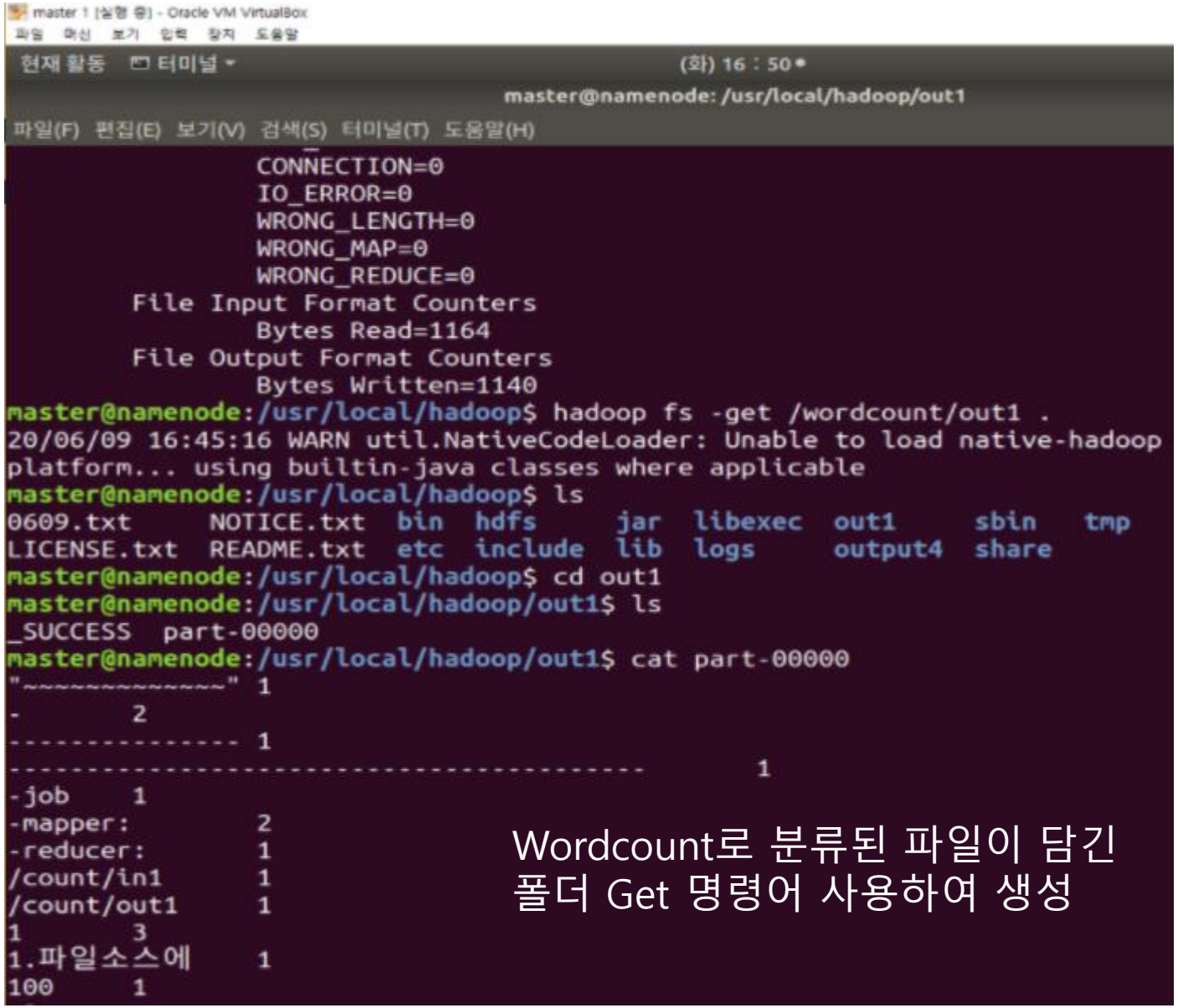
Hadoop wordcount – 실행결과

```
ubuntu@ubuntu-VirtualBox:/usr/local/hadoop$ hadoop jar jar/WordCount2.jar com.mega
.wc.WordCount2 /wordcount/input2 /wordcount/ouput2
        File System Counters
                FILE: Number of bytes read=394628
                FILE: Number of bytes written=1528452
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=212420
                HDFS: Number of bytes written=24360
                HDFS: Number of read operations=13
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
ubuntu@ubuntu-VirtualBox:/usr/local/hadoop$ hadoop fs -ls /wordcount/ouput2
21/06/02 13:13:02 WARN util.NativeCodeLoader: Unable to load native-hadoop library
 for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r--   1 ubuntu supergroup          0 2021-06-02 13:09 /wordcount/ouput2/_SU
CCESS
-rw-r--r--   1 ubuntu supergroup      24360 2021-06-02 13:09 /wordcount/ouput2/par
t-00000
ubuntu@ubuntu-VirtualBox:/usr/local/hadoop$ hadoop fs -cat /wordcount/ouput2/_SUCC
ESS
21/06/02 13:13:32 WARN util.NativeCodeLoader: Unable to load native-hadoop library
 for your platform... using builtin-java classes where applicable
ubuntu@ubuntu-VirtualBox:/usr/local/hadoop$ hadoop fs -cat /wordcount/ouput2/part-
00000
"printed         1
"program"        1
"recipient"      1
"screenplay      1
"software"),     7
"source"         1
"submitted"      1
"work"   2
"you"    2
"your") 1
'originates'     1
("agreement").   1
```

Hadoop jar jar/WordCount2.jar
com.mega.wc.WordCount2
/wordcount/input2
/wordcount/ouput2
WordCount.jar 파일로
wordcount 실행 결과 확인

Hadoop fs –cat
/wordcount/ouput2/part-00000
명령어 실행하여
최종 결과 확인

Hadoop wordcount – 실행결과



Wordcount로 분류된 파일이 담긴 폴더 Get 명령어 사용하여 생성

Window에서 txt 파일 csv 형태로 변경하기 위해 정제(특수기호, 불필요한 단어 삭제)

```
select count(*) from hadoop.counting where countint >= 2

select name from hadoop.counting where countint >= 2

select * from hadoop.counting where countint >= 2

select * from hadoop.counting where countint >= 2
order by countint desc limit 10
```

counting 1

select * from hadoop.counting where countint >= 2 Enter a SQL expression

	countint	name
1	12	the
2	6	and
3	5	of
4	4	software
5	4	this
6	4	for
7	3	export
8	3	encryption
9	3	cryptographic
10	2	security

csv 파일 DB에 넣고 sql 문으로 order by 사용하여 단어빈도 확인

Hadoop wordcount – 실행결과

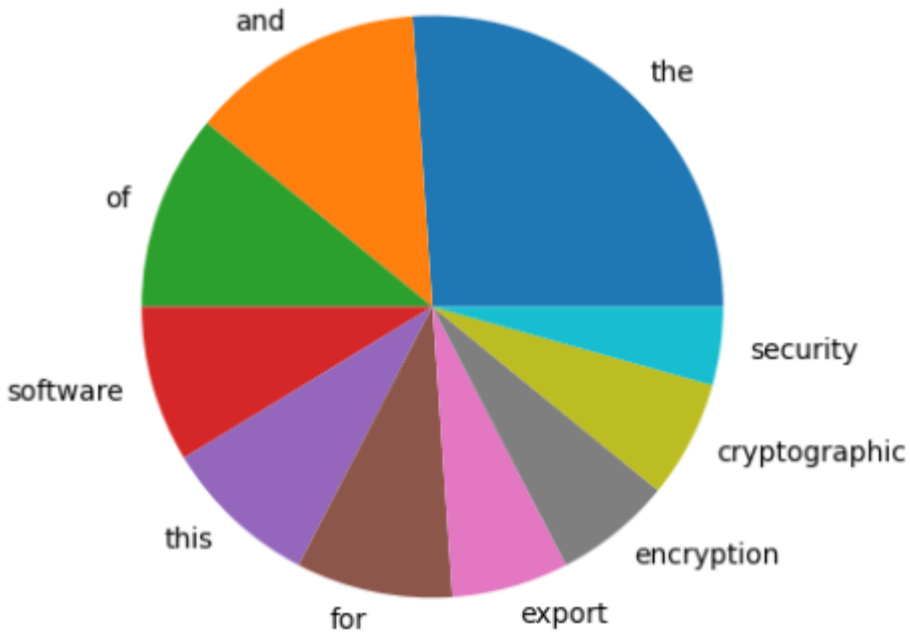
```python
import matplotlib.pyplot as plt
import pymysql

def select_all():
    conn = pymysql.connect(
        user='root',
        passwd='1234',
        host='localhost',
        port=3306,
        db='hadoop',
        charset='utf8'
    )
    # print(conn.db)
    # cursor = conn.cursor(pymysql.cursors.DictCursor)  # 결과값을 딕셔너리 형태로 가져올 경우
    cursor = conn.cursor()  # 결과값을 튜플 형태로 가져올 경우
    sql = "select * from hadoop.counting where countint >= 2 order by countint desc limit 10"
    cursor.execute(sql, )
    conn.commit()
    result = cursor.fetchall()
    print(result)
    # 튜플 형태를 df 시리즈 형태로(데이터시각화 하기 위해서)
    countint = list()
    name = list()


    # name, countint 컬럼에 값을 넣어줌
    for x, y in result:
        print(x, y)
        name.append(y)
        countint.append(x)
    print(name)
    print(countint)

    plt.figure()
    plt.pie(countint, labels=name)
    plt.show()

if __name__ == '__main__':
    select_all()
```

mySQL을 파이썬에서 사용하기 위한 pymysql 및 시각화를 위해 Matplotlib 라이브러리 설치

Sql 연결 → 결과값(csv) 튜플 형태로 가져옴

→ order by 문 사용하여 단어 빈도가 높은 단어 확인

→ 시각화 하기 위해 튜플 형태의 데이터를 시리즈 형태로 변환하기 위해 for 문 사용하고 name, countint 변수에 값들을 더해줌 → 파이 차트로 시각화



HiveQL – 설치 및 연결

apache-hive 다운로드

```
ubuntu@ubuntu-VirtualBox:~$ wget http://mirror.navercorp.com/apache
8/apache-hive-2.3.8-bin.tar.gz
--2021-06-03 11:48:23--  http://mirror.navercorp.com/apache/hive/hi
-hive-2.3.8-bin.tar.gz
mirror.navercorp.com (mirror.navercorp.com)을(를) 해석하는 중... 12
접속 mirror.navercorp.com (mirror.navercorp.com)|125.209.216.167|:8
HTTP 요청을 전송했습니다. 응답을 기다리는 중입니다... 200 OK
길이: 286137131 (273M) [application/octet-stream]
다음 위치에 저장: `apache-hive-2.3.8-bin.tar.gz'

.3.8-bin.tar.gz       56%[==========>          ] 155.19M  35.5MB/s
```

환경변수 설정

```
ubuntu@ubuntu-VirtualBox:~$ cd apache-hive-2.3.8/
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8$ pwd
/home/ubuntu/apache-hive-2.3.8
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8$ cd ..
ubuntu@ubuntu-VirtualBox:~$ sudo gedit .bashrc
[sudo] ubuntu의 암호:
```

관리자 권한으로 source .bashrc 접속하여
환경변수 설정 실행

```
16 ############ HIVE ###############
17 export HIVE_HOME=/home/ubuntu/apache-hive-2.3.8
18 export PATH=$PATH:$HIVE_HOME/bin
19
```

Hadoop - Hive 연결

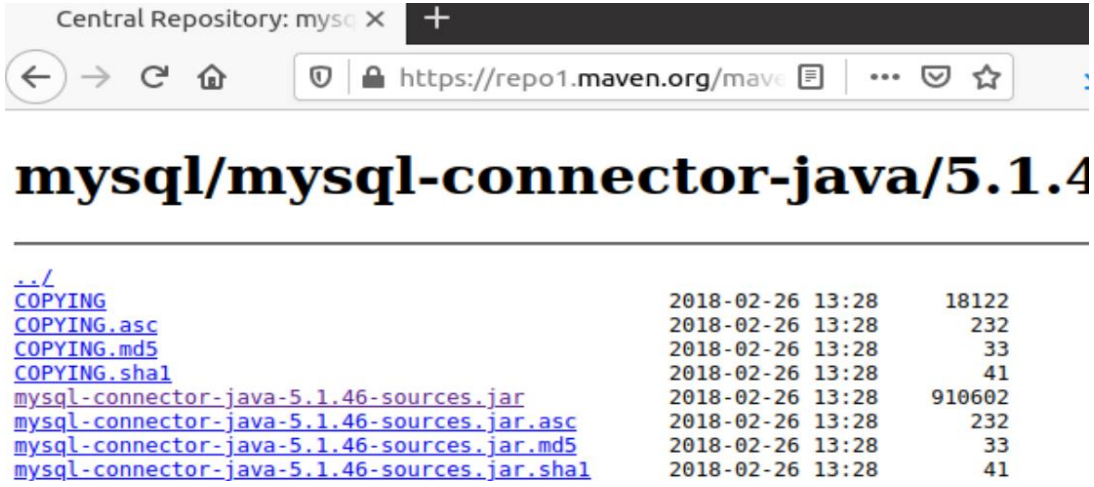
```
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8$ cd conf/
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/conf$ ls
beeline-log4j2.properties.template    ivysettings.xml
hive-default.xml.template             llap-cli-log4j2.properties.template
hive-env.sh.template                  llap-daemon-log4j2.properties.template
hive-exec-log4j2.properties.template  parquet-logging.properties
hive-log4j2.properties.template
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/conf$ cp hive-env.sh.template h
v.sh
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/conf$ ls
beeline-log4j2.properties.template    hive-log4j2.properties.template
hive-default.xml.template             ivysettings.xml
hive-env.sh                           llap-cli-log4j2.properties.template
hive-env.sh.template                  llap-daemon-log4j2.properties.template
hive-exec-log4j2.properties.template  parquet-logging.properties
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/conf$ sudo gedit hive-env.sh
[sudo] ubuntu의 암호:
```

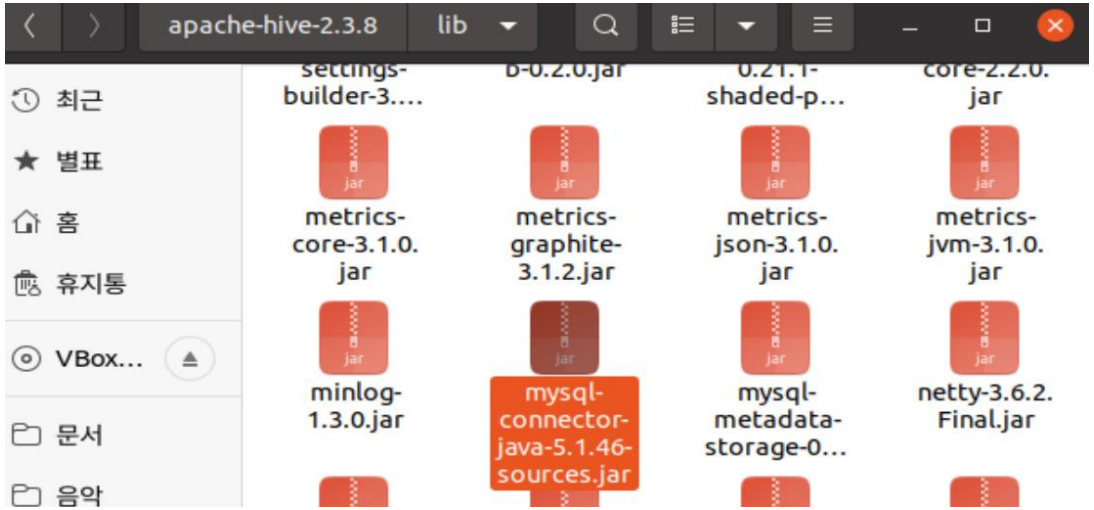
관리자 권한으로 hive-env 접속하여
HAOOP_HOME 설치 된 주소 입력

```
4 # appropriate for hive server.
5
6
7 # Set HADOOP_HOME to point to a specific hadoop install directory
8 HADOOP_HOME=/usr/local/hadoop
9
0 # Hive Configuration Directory can be controlled by:
1 # export HIVE_CONF_DIR=
2
```

HiveQL – 설치 및 연결

Mysql 다운로드 및 hive /lib 폴더에 저장
hive = java, jdbc 연결하기 위함





mysql 접속하기 위한 계정연결

```
ubuntu@ubuntu-VirtualBox:~$ cd apache-hive-2.3.8/conf
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/conf$ ls
beeline-log4j2.properties.template    hive-log4j2.properties.template
hive-default.xml.template             ivysettings.xml
hive-env.sh                           llap-cli-log4j2.properties.template
hive-env.sh.template                  llap-daemon-log4j2.properties.template
hive-exec-log4j2.properties.template  parquet-logging.properties
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/conf$ sudo gedit hive-site.xml
```

열기(O) hive-site.xml /home/ubuntu/apache-hive-2.3.8/conf 저장(S)

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3 <configuration>
4     <property>
5         <name>hive.metastore.local</name>
6         <value>true</value>
7     </property>
8     <property>
9         <name>javax.jdo.option.ConnectionURL</name>
10            <value>jdbc:mysql://localhost:3306/hive?-
  useSSL=false&amp;createDatabaseIfNotExist=true</value>
11    </property>
12    <property>
13        <name>javax.jdo.option.ConnectionDriverName</name>
14        <value>com.mysql.jdbc.Driver</value>
15    </property>
16    <property>
17        <name>javax.jdo.option.ConnectionUserName</name>
18        <value>hive</value>
19    </property>
20    <property>
21        <name>javax.jdo.option.ConnectionPassword</name>
22        <value>hive</value>
23    </property>
24 </configuration>
```

HiveQL – 설치 및 연결

저장소 목록과 리눅스 목록 맞춰주는 업데이트 진행

```
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/conf$ sudo apt-get update
[sudo] ubuntu의 암호:
받기:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
기존:2 http://kr.archive.ubuntu.com/ubuntu focal InRelease
받기:3 http://kr.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
받기:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages
```

Mysql 서버 설치 진행

```
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/conf$ sudo apt-get install mysql-serv
er
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음의 추가 패키지가 설치될 것입니다 :
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
  mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-server-8.0 mysql-server-core-8.0
제안하는 패키지:
  libipc-sharedcache-perl mailx tinyca
다음 새 패키지를 설치할 것입니다:
```

방화벽 세부 설정

```
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/conf$ sudo ufw allow mysql
규칙이 업데이트됐습니다
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/conf$ sudo systemctl enable start
Failed to enable unit: Unit file start.service does not exist.
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/conf$ sudo systemctl start mysql
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/conf$ sudo systemctl enable start
Failed to enable unit: Unit file start.service does not exist.
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/conf$ sudo systemctl enable mysql
Synchronizing state of mysql.service with SysV service script with /lib/systemd
/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mysql
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/conf$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
     Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset>
```

mysql 원격 접속

```
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/conf$ sudo /usr/bin/mysql -u root
-p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.25-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

모든 곳에서 접속할 수 있는 root 사용자 생성
Root 에게 모든 DB, 모든 data에 접근 가능한 권한 부여
권한 부여 실행

```
mysql> create user 'hive'@'%' identified by 'hive';
Query OK, 0 rows affected (0.01 sec)

mysql> grant all privileges on *.* to 'hive'@'%' with grant option;
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)
```

HiveQL – hiveQL 실행

하둡 namenode 포맷 및 Hadoop 실행

```
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/bin$ hadoop namenode -format
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
```

```
ubuntu@ubuntu-VirtualBox:~/apache-hive-2.3.8/bin$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
21/06/03 13:05:33 WARN util.NativeCodeLoader: Unable to load native-hadoop
 for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
ubuntu@localhost's password:
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-ubu
node-ubuntu-VirtualBox.out
```

```
hive> create table product (
    > id string,
    > title string,
    > content string,
    > price string
    > );
```

테이블 생성 및 insert문 사용하여 입력 값 넣기

```
hive> insert into product values ('100', 'car', 'fun car', 1000);
hive> insert into product values ('300', 'car2', 'fun car2', 1000);
hive> insert into product values ('200', 'car2', 'fun car2', 2000);
```

```
hive> select * from product;
OK
100     car     fun car 1000
200     car2    fun car2        2000
300     car2    fun car2        1000
Time taken: 0.374 seconds, Fetched: 3 row(s)
```

Insert director문 사용하여 Product 테이블의 데이터를 읽어서 지정한 위치에서 파일을 출력

* Hive는 HDFS 파일이 수정 불가함으로 UPDATE, DELETE 사용 불가하기 때문에 insert overwrite 키워드를 사용

```
hive> insert overwrite local directory '/home/hadoop/data3'
    > select * from product where title = 'car2'
    > ;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the fu
re versions. Consider using a different execution engine (i.e. spark, tez) or
ing Hive 1.X releases.
Query ID = hadoop_20210603165405_650e3410-8c45-41ad-a8b3-0cbb5cc52c83
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
2021-06-03 16:54:07,450 Stage-1 map = 100%,  reduce = 0%
Ended Job = job_local1524668322_0007
Moving data to local directory /home/hadoop/data3
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 735 HDFS Write: 601 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 1.626 seconds
hive> insert overwrite local directory '/home/hadoop/data4'
    > select * from product where price = '1000';
```

```
hadoop@hadoop:~$ cd ~/data
hadoop@hadoop:~/data$ cat 000000_0
apple□apple□apple□apple
apple2□apple2□apple2□apple2
hadoop@hadoop:~/data$ cd ..
hadoop@hadoop:~$ cd ~/data3
hadoop@hadoop:~/data3$ cat 000000_0
200□car2□fun car2□2000
hadoop@hadoop:~/data3$ cd ..
hadoop@hadoop:~$ cd ~/data4
hadoop@hadoop:~/data4$ cat 000000_0
100□car□fun car□1000
hadoop@hadoop:~/data4$
```

터미널에 들어가 지정한 파일 위치에서 데이터 읽어 오기

HiveQL – join 함수 사용

```
hive> create table member (
    > id string,
    > pw string,
    > name string,
    > tel string
    > )
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 0.234 seconds
```

테이블 생성 및 insert문 사용하여 입력 값 넣기

Row format은 필드의 쉼표를 제거하고 데이터를 컬럼단위로 구분함

member.csv 데이터를 load하여 member table 에 overwrite한 후, select 문 사용하여 확인

```
hive> load data local inpath '/home/hadoop/data/member.csv'
    > overwrite into table member;
Loading data to table default.member
OK
Time taken: 0.754 seconds
hive> set hive.cli.print.header=true;
```

```
hive> select * from member;
OK
member.id       member.pw       member.name     member.tel
park    park    park    011
song    song    park    011
jung    jung    jung    012
apple   apple   apple   012
id5     pw      apple   012
id6     song    apple   013
id7     song    name    013
id8     pw      name    013
id8     pw      name    013
Time taken: 0.24 seconds, Fetched: 9 row(s)
```

bbs2.csv 데이터 load 후, bbs2 테이블에 overwrite 한 후, select 문 사용하여 확인

```
hive> select * from bbs2;
OK
bbs2.id bbs2.title      bbs2.content    bbs2.writer     bbs2.prepost
110     title11 fun title01     park    post
220     title12 fun title02     park    post
330     title13 fun title03     park    post
440     title14 fun title04     song    post
550     title15 fun title05     song    post
660     title16 fun title06     song    post
770     title17 fun title07     jung    post
880     title18 fun title08     jung    post
```

```
member.id       member.pw       member.name     member.tel
park    park    park    011
song    song    park    011
jung    jung    jung    012
apple   apple   apple   012
id5     pw      apple   012
id6     song    apple   013
id7     song    name    013
id8     pw      name    013
id8     pw      name    013
```

```
hive> select m.name, b.title from member m
    > join bbs b
    > on m.id = b.writer;
```

member 테이블의 id와 bbs 테이블의 writer가 동일한 값의 m.name, m.title 을 bbs 테이블에 입력

```
m.name  b.title
park    title01
park    title02
park    title03
park    title04
park    title05
park    title06
jung    title07
jung    title08
```