데이터과학
10주차
HIVE

201202156
오희빈

1. HIVE 설치 및 실행

```
hadoop01@hadoop00:~$ sudo wget http://apache.mirror.cdnetworks.com/hive/hive-1.2
.2/apache-hive-1.2.2-bin.tar.gz
```

=> wget을 통해 hive를 다운로드 받는다.

```
hadoop01@hadoop00:~$ ls
apache-hive-1.2.2-bin          examples.desktop        patterns       testfile1
apache-hive-1.2.2-bin.tar.gz   hadoop-2.7.3            patterns.txt   testfile2
derby.log                      hadoop-2.7.3.tar.gz     Pictures       Videos
Desktop                        jdk-7u80-linux-x64.rpm  Public
Documents                      metastore_db            tashu.csv
Downloads                      Music                   Templates
```

=> hive.tar.gz 압축 파일의 압축을 풀어 준다.

```
hadoop01@hadoop00:~$ export HIVE_HOME=/home/hadoop01/apache-hive-1.2.2-bin
```

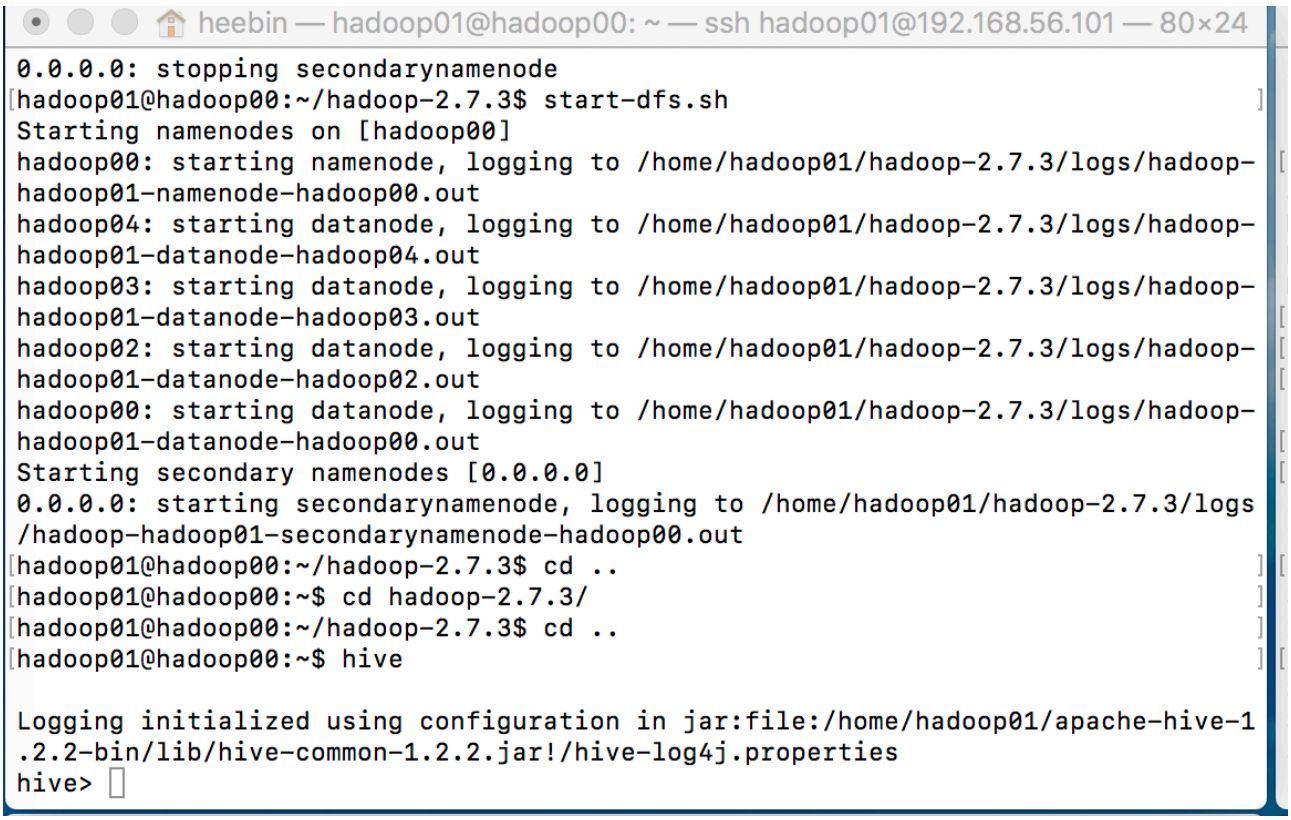
=> HIVE_HOME의 환경변수를 HIVE의 경로로 export 한다.

```
hadoop01@hadoop00:~$ export PATH=$HIVE_HOME/bin:$PATH
```

=> PATH 환경변수에 HIVE_HOME/bin을 추가로 export 한다.

```
hadoop01@hadoop03:~/hadoop-2.7.3$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /home/hadoop01/hadoop-2.7.3/logs/yarn-hadoo
p01-resourcemanager-hadoop03.out
hadoop02: starting nodemanager, logging to /home/hadoop01/hadoop-2.7.3/logs/yarn
-hadoop01-nodemanager-hadoop02.out
hadoop00: starting nodemanager, logging to /home/hadoop01/hadoop-2.7.3/logs/yarn
-hadoop01-nodemanager-hadoop00.out
hadoop04: starting nodemanager, logging to /home/hadoop01/hadoop-2.7.3/logs/yarn
-hadoop01-nodemanager-hadoop04.out
hadoop03: starting nodemanager, logging to /home/hadoop01/hadoop-2.7.3/logs/yarn
-hadoop01-nodemanager-hadoop03.out
hadoop01@hadoop03:~/hadoop-2.7.3$ jps
4322 NodeManager
4054 DataNode
4359 Jps
4183 ResourceManager
hadoop01@hadoop03:~/hadoop-2.7.3$
```

=> 클러스터 4개를 모두 사용해서 hive를 이용. start-yarn.sh 를 해서 nodemanager와 resourcemanager를 실행시켜준다.



=> start-dfs.sh 를 실행시켜 namenode와 datanode, secondarynamenode를 실행 시켜 준다.
그리고 HIVE를 실행 시켜준다.

2. 타슈 데이터 HIVE에 업로드하기

```
ohheebinui-MacBook-Pro:Desktop heebin$ rsync -avz /Users/heebin/Desktop/tashu.csv hadoop01@192.168.56.101:/home/hadoop01/
hadoop01@192.168.56.101's password:
building file list ... done
tashu.csv

sent 28897936 bytes  received 42 bytes  2512867.65 bytes/sec
total size is 124163945  speedup is 4.30
ohheebinui-MacBook-Pro:Desktop heebin$ rsync -avz /Users/heebin/Desktop/tashu.csv hadoop01@192.168.56.102:/home/hadoop01/
hadoop01@192.168.56.102's password:
building file list ... done
tashu.csv

sent 28897936 bytes  received 42 bytes  2512867.65 bytes/sec
total size is 124163945  speedup is 4.30
ohheebinui-MacBook-Pro:Desktop heebin$ rsync -avz /Users/heebin/Desktop/tashu.csv hadoop01@192.168.56.103:/home/hadoop01/
hadoop01@192.168.56.103's password:
building file list ... done
tashu.csv

sent 28897936 bytes  received 42 bytes  2311838.24 bytes/sec
total size is 124163945  speedup is 4.30
ohheebinui-MacBook-Pro:Desktop heebin$ rsync -avz /Users/heebin/Desktop/tashu.csv hadoop01@192.168.56.104:/home/hadoop01/
hadoop01@192.168.56.104's password:
building file list ... done
tashu.csv

sent 28897936 bytes  received 42 bytes  2752188.38 bytes/sec
total size is 124163945  speedup is 4.30
ohheebinui-MacBook-Pro:Desktop heebin$
```

=> 기존 컴퓨터에 존재하는 tashu.csv 파일을 rsync를 이용해서 4개의 클러스터에 모두 옮겨 준다.

```
hive> create table tmp (RENT_STATION int, RENT_DATE string, RETURN_STATION int, RETURN_DATE string) row format delimited fields terminated by ',';
OK
Time taken: 0.495 seconds
hive> create table tashu (RENT_STATION int, RENT_DATE timestamp, RETURN_STATION int, RETURN_DATE timestamp) row format delimited fields terminated by ',';
OK
Time taken: 0.119 seconds
hive> load data local inpath '/home/hadoop01/tashu.csv' overwrite into table tmp
;
Loading data to table default.tmp
Table default.tmp stats: [numFiles=1, numRows=0, totalSize=124163945, rawDataSize=0]
OK
Time taken: 4.208 seconds
```

=> RENT_DATE와 RETURN_DATE의 date타입의 데이터베이스를 저장하기 위해 RENT_DATE와 RETURN_DATE를 string으로 가지고 있는 tmp table을 하나 그리고 RENT_DATE와 RETURN_DATE를

date 타입으로 저장하기 위한 tashu table을 생성한다. 그리고 tmp table에 tashu.csv를 overwrite시켜 디비에 저장 시킨다.

```
hive> insert into table tashu
    > select RENT_STATION,
    > from_unixtime(unix_timestamp(RENT_DATE, 'yyyyMMddHHmmss')),
    > RETURN_STATION,
    > from_unixtime(unix_timestamp(RETURN_DATE, 'yyyyMMddHHmmss'))
    > from tmp;
Query ID = hadoop01_20170601200338_660bbf6e-6672-4df5-8dd6-7bae688c5490
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1496311355444_0001, Tracking URL = http://hadoop03:8088/proxy
/application_1496311355444_0001/
Kill Command = /home/hadoop01/hadoop-2.7.3/bin/hadoop job  -kill job_14963113554
44_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-06-01 20:03:55,034 Stage-1 map = 0%,  reduce = 0%
2017-06-01 20:04:56,007 Stage-1 map = 0%,  reduce = 0%, Cumulative CPU 38.13 sec
2017-06-01 20:05:06,756 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 49.67 s
ec
MapReduce Total cumulative CPU time: 49 seconds 670 msec
Ended Job = job_1496311355444_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://hadoop00/user/hive/warehouse/tashu/.hive-staging_hive_201
7-06-01_20-03-38_015_5800191555922286709-1/-ext-10000
Loading data to table default.tashu
Table default.tashu stats: [numFiles=1, numRows=3404664, totalSize=157063263, ra
wDataSize=153658599]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 49.67 sec   HDFS Read: 124168047 HDFS Wr
ite: 157063344 SUCCESS
Total MapReduce CPU Time Spent: 49 seconds 670 msec
OK
Time taken: 91.496 seconds
hive>
```

=> tashu table에 RENT_DATE와 RETURN_DATE를 date 타입으로 저장 하기 위해서 unix_timestamp(RENT_DATE, 'yyyyMMddHHmmss')와 unix_timestamp(RETURN_DATE, 'yyyyMMddHHmmss') 를 이용해서 tashu table에 다시 저장 시킨다. 이 과정에서 타입을 바꾸기 때문인지 mapreduce를 하지만 map만 실행시킨다.

```
hive> select * from tashu limit 10;
OK
NULL    NULL    NULL    NULL
43      2013-01-01 05:56:03     34      2013-01-01 06:02:17
97      2013-01-01 06:04:00     NULL    2013-01-01 10:20:37
2       2013-01-01 06:04:06     10      2013-01-01 06:18:59
106     2013-01-01 10:53:05     105     2013-01-01 10:57:43
4       2013-01-01 11:22:23     4       2013-01-01 12:17:53
21      2013-01-01 11:39:53     105     2013-01-01 11:49:43
90      2013-01-01 12:08:33     91      2013-01-01 12:51:36
13      2013-01-01 13:14:29     30      2013-01-01 13:30:39
1       2013-01-01 13:37:42     1       2013-01-01 13:38:15
Time taken: 0.153 seconds, Fetched: 10 row(s)
```

=> tashu table에 저장된 내용을 확인 limit 10;을 통해서 10개의 row만 보여주며 date 타입으로 저장된 것을 확인 할 수 있다.

3. 타슈 데이터에 쿼리를 날려 통계 구하기

1) 연도별

```
hive> select year(RENT_DATE),count(*) as cnt from tashu group by year(RENT_DATE)
;
Query ID = hadoop01_20170601201547_886db48b-cefd-4503-81b8-cc650a3d12f7
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1496311355444_0004, Tracking URL = http://hadoop03:8088/proxy
/application_1496311355444_0004/
Kill Command = /home/hadoop01/hadoop-2.7.3/bin/hadoop job  -kill job_14963113554
44_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-06-01 20:15:59,892 Stage-1 map = 0%,  reduce = 0%
2017-06-01 20:16:15,470 Stage-1 map = 57%,  reduce = 0%, Cumulative CPU 8.46 sec
2017-06-01 20:16:16,510 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.41 se
c
2017-06-01 20:16:23,802 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 10.84
 sec
MapReduce Total cumulative CPU time: 10 seconds 840 msec
Ended Job = job_1496311355444_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 10.84 sec   HDFS Read: 157075
302 HDFS Write: 44 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 840 msec
OK
NULL    1
2013    1036614
2014    1200187
2015    1167862
Time taken: 38.587 seconds, Fetched: 4 row(s)
```

=> select year(RENT_DATE), count(*) as cnt from tashu group by year(RENT_DATE); 의 쿼리문을 이용해서 date 타입으로 저장된 RENT_DATE의 year를 group by 를 통해서 같은 연도별로 묶어 주고 count 하게 한다.

2) 월별

```
hive> select month(RENT_DATE),count(*) as cnt from tashu group by month(RENT_DAT
E);
Query ID = hadoop01_20170601201739_51163896-47fe-40f9-9fb3-b540515bf6d4
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1496311355444_0005, Tracking URL = http://hadoop03:8088/proxy
/application_1496311355444_0005/
Kill Command = /home/hadoop01/hadoop-2.7.3/bin/hadoop job  -kill job_14963113554
44_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-06-01 20:17:54,227 Stage-1 map = 0%,  reduce = 0%
2017-06-01 20:18:13,248 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 8.99 se
c
2017-06-01 20:18:23,718 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 10.83
 sec
MapReduce Total cumulative CPU time: 10 seconds 830 msec
Ended Job = job_1496311355444_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 10.83 sec   HDFS Read: 157075
311 HDFS Write: 115 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 830 msec
OK
NULL    1
1       99693
2       116987
3       232712
4       290519
5       414934
6       480429
7       358507
8       320058
9       366859
10      360240
11      214771
12      148954
Time taken: 46.168 seconds, Fetched: 13 row(s)
```

=> select month(RENT_DATE), count(*) as cnt from tashu group by month(RENT_DATE); 의 쿼리문을 이용해서 date 타입으로 저장된 RENT_DATE의 month를 group by 를 통해서 같은 월별로 묶어 주고 count 하게 한다.

3) 일별

```
hive> select day(RENT_DATE),count(*) as cnt from tashu group by day(RENT_DATE);
Query ID = hadoop01_20170601201938_25689018-4ee1-48d3-b116-8bfa57189644
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1496311355444_0006, Tracking URL = http://hadoop03:8088/proxy
/application_1496311355444_0006/
Kill Command = /home/hadoop01/hadoop-2.7.3/bin/hadoop job  -kill job_14963113554
44_0006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-06-01 20:19:48,949 Stage-1 map = 0%,  reduce = 0%
2017-06-01 20:20:03,415 Stage-1 map = 57%,  reduce = 0%, Cumulative CPU 8.36 sec
2017-06-01 20:20:04,445 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.22 se
c
2017-06-01 20:20:13,745 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 10.94
 sec
MapReduce Total cumulative CPU time: 10 seconds 940 msec
Ended Job = job_1496311355444_0006
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 10.94 sec   HDFS Read: 157075
307 HDFS Write: 305 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 940 msec
OK
NULL    1
1       116298
2       104499
3       106474
4       115649
5       115198
6       110271
7       105940
8       107095
9       117166
10      110197
11      105053
12      104176
13      107960
14      110600
15      120347
16      120099
17      114330
18      105077
19      112020
20      106850
21      113472
22      120796
23      111161
24      110613
25      104908
26      120407
27      106867
28      119053
29      102871
30      114542
31      64674
Time taken: 36.218 seconds, Fetched: 32 row(s)
```

=> select day(RENT_DATE), count(*) as cnt from tashu group by day(RENT_DATE); 의 쿼리문을 이용해서 date 타입으로 저장된 RENT_DATE의 day를 group by 를 통해서 같은 일별로 묶어 주고 count 하게 한다.

4) 시간별

```
hive> select hour(RENT_DATE),count(*) as cnt from tashu group by hour(RENT_DATE)
;
Query ID = hadoop01_20170601202044_14beddca-005a-4b0f-b194-7e3c40e6f12f
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1496311355444_0007, Tracking URL = http://hadoop03:8088/proxy
/application_1496311355444_0007/
Kill Command = /home/hadoop01/hadoop-2.7.3/bin/hadoop job  -kill job_14963113554
44_0007
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-06-01 20:20:54,583 Stage-1 map = 0%,  reduce = 0%
2017-06-01 20:21:10,040 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 8.92 se
c
2017-06-01 20:21:20,358 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 10.73
 sec
MapReduce Total cumulative CPU time: 10 seconds 730 msec
Ended Job = job_1496311355444_0007
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 10.73 sec   HDFS Read: 157075
302 HDFS Write: 218 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 730 msec
OK
NULL    1
0       63022
1       15205
2       199
3       3
4       11
5       16591
6       22152
7       129110
8       186421
9       142126
10      111652
11      96250
12      118768
13      157488
14      173437
15      167018
16      194413
17      250842
18      292905
19      241023
20      259816
21      273561
22      251385
23      241265
Time taken: 37.23 seconds, Fetched: 25 row(s)
```

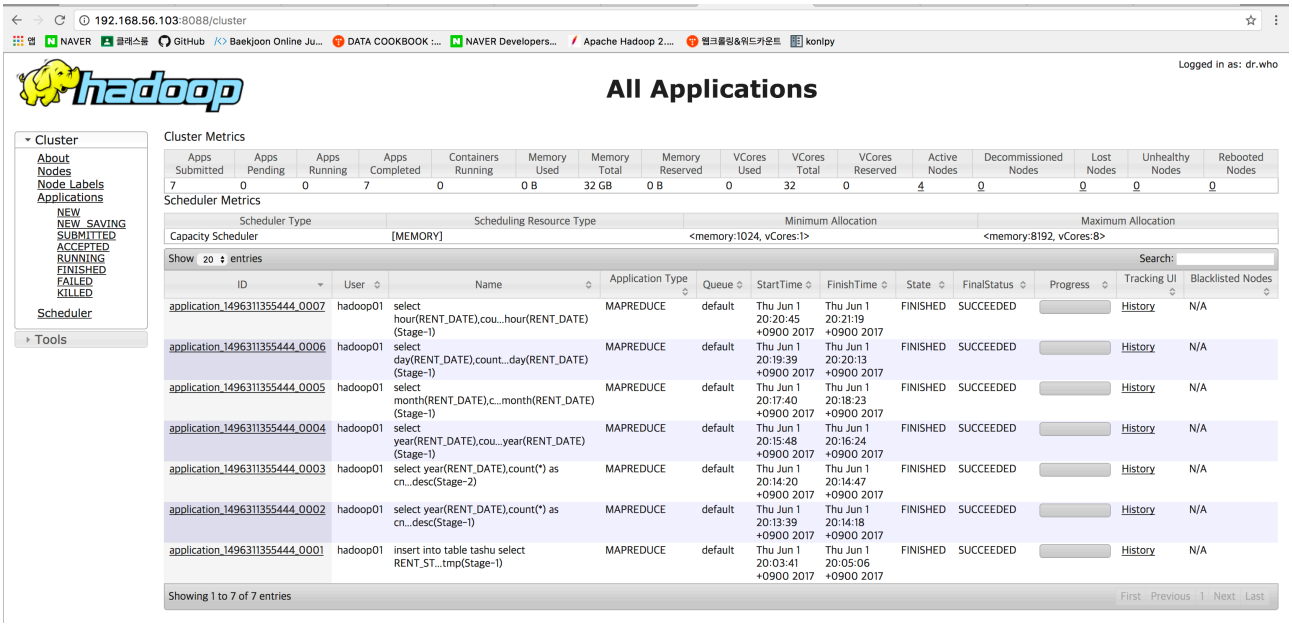
=> select hour(RENT_DATE), count(*) as cnt from tashu group by hour(RENT_DATE); 의 쿼리문을 이용해서 date 타입으로 저장된 RENT_DATE의 hour를 group by 를 통해서 같은 시간별로 묶어 주고 count 하게 한다.

```
hadoop01@hadoop00:~/hadoop-2.7.3$ hdfs dfsadmin -report
Configured Capacity: 81369726976 (75.78 GB)
Present Capacity: 53813940224 (50.12 GB)
DFS Remaining: 52671918080 (49.05 GB)
DFS Used: 1142022144 (1.06 GB)
DFS Used%: 2.12%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

-------------------------------------------------
Live datanodes (4):

Name: 192.168.56.101:50010 (hadoop00)
Hostname: hadoop00
Decommission Status : Normal
Configured Capacity: 20342431744 (18.95 GB)
DFS Used: 285511680 (272.29 MB)
Non DFS Used: 6925783040 (6.45 GB)
DFS Remaining: 13131137024 (12.23 GB)
DFS Used%: 1.40%
DFS Remaining%: 64.55%
Configured Cache Capacity: 0 (0 B)
```

hdfs dfsadmin -report

```
hadoop01@hadoop03:~$ yarn node -list
17/06/01 20:43:08 INFO client.RMProxy: Connecting to ResourceManager at hadoop03
/192.168.56.103:8032
Total Nodes:4
         Node-Id             Node-State Node-Http-Address       Number-of-Runnin
g-Containers
  hadoop03:42230                RUNNING     hadoop03:8042
            0
  hadoop02:42874                RUNNING     hadoop02:8042
            0
  hadoop00:38715                RUNNING     hadoop00:8042
            0
  hadoop04:41044                RUNNING     hadoop04:8042
            0
hadoop01@hadoop03:~$
```

yarn node -list



=> 192.168.56.103:8088 을 통해서 mapreduce 결과를 확인 할 수 있다.