□ 수행평가 - 빅데이터 분석시스템 구축 과정

1. Hadoop을 설치 한다.

- hadoop과 j에 압축 풀기

```
[root@master 다운로드]# tar xvfz hadoop-1.2.1.tar.gz
```

```
[root@master 다운로드]# tar xvfz jdk-8u161-linux-x64.tar.gz
```

- usr/local 에 압축 푼 파일 복사₩

```
[root@master 다운로드]# cp -r jdk1.8.0_161/ /usr/local
[root@master 다운로드]# cp -r hadoop-1.2.1 /usr/local
```

2. 설정 파일을 설정 한다.

- 환경 설정 파일을 수정 및 소프트링크 연결 (jdk 확인 필수)

```
[root@master ~]# vi /etc/profile
```

```
50      export HISTCONTROL=ignoredups
51  fi
52  JAVA_HOME=/usr/local/jdk-1.8.0_161
53  HADOOP_HOME=/usr/local/hadoop-1.2.1
54  CLASSPATH=/usr/local/jdk-1.8.0_161/lib
55  export JAVA_HOME
56  export HADOOP_HOME
57  export CLASSPATH
58  PATH=$HADOOP_HOME/bin:$JAVA_HOME/bin:$PATH
59  export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE
60
```

```
[root@master bin]# ln -s /usr/local/jdk1.8.0_161/bin/java java
```

```
[root@master bin]# java -version
java version "1.8.0_161"
Java(TM) SE Runtime Environment (build 1.8.0_161-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)
[root@master bin]#
```

- ssh 설정 및 확인

```
[root@master ~]# ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
Generating public/private dsa key pair.
/root/.ssh/id_dsa already exists.
Overwrite (y/n)? y
Your identification has been saved in /root/.ssh/id_dsa.
Your public key has been saved in /root/.ssh/id_dsa.pub.
The key fingerprint is:
a3:de:58:56:11:b1:ce:ad:b0:98:85:6b:48:dc:71:7a root@master
The key's randomart image is:
+--[ DSA 1024]----+
|           o.    |
|            o    |
|       . . o     |
|    . .  = o o   |
|     o + E + .   |
|    . .  B = .   |
|     . * + .     |
|      o =        |
|       o .       |
+-----------------+
[root@master ~]# cd .ssh
[root@master .ssh]# ls
authrized_keys  id_dsa  id_dsa.pub  known_hosts
[root@master .ssh]#
```

Authorized_keys

```
[root@master .ssh]# cat id_dsa.pub >> authrized_keys
[root@master .ssh]# ls
authrized_keys  id_dsa  id_dsa.pub  known_hosts
```

- core-site.xml, hdfs-site.xml, mapred-site.xml 을 차례로 아래와 같이 기입

```
[root@master ~]# cd /usr/local/hadoop-1.2.1/conf
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
<name>fs.default.name</name>
<value>hdfs://192.168.111.101:9000</value>
</property>
<property>
<name>dfs.tmp.dir</name>
<value>/usr/local/hadoop-1.2.1/tmp</value>
</property>

</configuration>
~
~
```

```
<?xmite-specific property overrides in this file. -->

<configuration>

<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.http.address</name>
<value>192.168.111.101:50070</value>
</property>
<property>
<name>dfs.name.dir</name>
<value>/usr/local/hadoop-1.2.1/name</value>
</property>
<property>
<name>dfs.data.dir</name>
<value>/usr/local/hadoop-1.2.1/data</value>
</property>
<property>
<name>dfs.webhdfs.enabled</name>
<value>true</value>
</property>

</configuration>
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>mapred.job.tracker</name>
<value>192.168.111.101:9001</value>
</property>
</configuration>
~
~
```

- 방화벽 해제

```
[root@master conf]# systemctl stop firewalld
[root@master conf]# systemctl disable firewalld
```

- usr/local/hadoop-1.2.1/conf 안의 hadoop-env.sh를 아래와 같이 추가 수정

```
# The java implementation to use.  Required.
 export JAVA_HOME=/usr/lib/jdk1.8.0_161
 export HADOOP_HOME_WARN_SUPPRESS="TRUE"
```

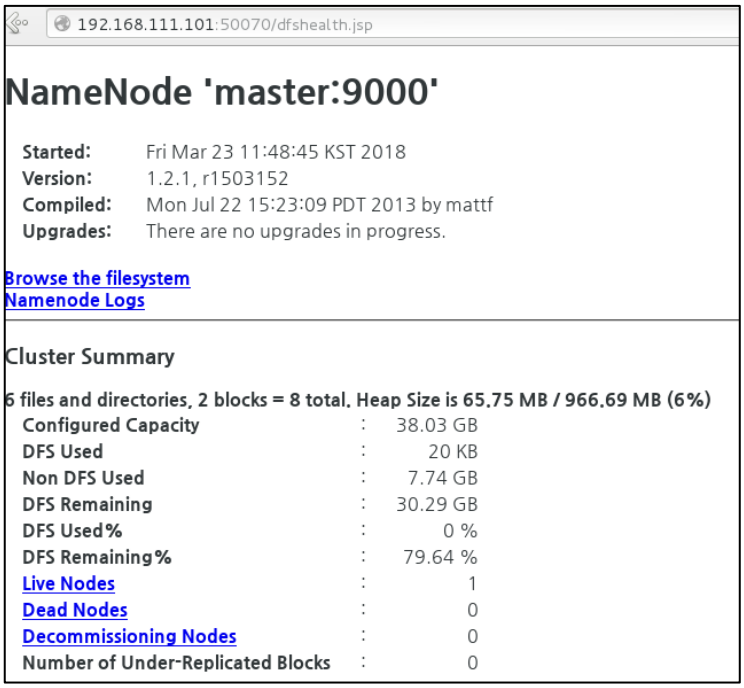
3. Hadoop을 가동 한다.

- hadoop namenode –format 후, 리부팅. hadoop 정상 동작 확인

```
[root@master ~]# hadoop
/usr/local/hadoop-1.2.1/libexec/../conf/hadoop-env.sh: line 8: The: command not
found
Usage: hadoop [--config confdir] COMMAND
where COMMAND is one of:
  namenode -format     format the DFS filesystem
  secondarynamenode    run the DFS secondary namenode
  namenode             run the DFS namenode
  datanode             run a DFS datanode
  dfsadmin             run a DFS admin client
  mradmin              run a Map-Reduce admin client
  fsck                 run a DFS filesystem checking utility
  fs                   run a generic filesystem user client
  balancer             run a cluster balancing utility
```

4. Hadoop 관리화면 가동

- hadoop 관리화면 가동



5. Hive 설치

- MariaDB 관련 파일 설치

```
[root@master ~]# cd 다운로드
[root@master 다운로드]# yum -y localinstall Maria*
```

- profile에 Hive 환경설정 추가

```
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL
JAVA_HOME=/usr/local/jdk1.8.0_161
HADOOP_HOME=/usr/local/hadoop-1.2.1
HIVE_HOME=/usr/local/hive
CLASSPATH=/usr/local/jdk1.8.0_161/lib
export JAVA_HOME HADOOP_HOME HIVE_HOME CLASSPATH
PATH=$HADOOP_HOME/bin:$JAVA_HOME/bin:$HIVE_HOME/bin:$PATH
```

- apache hive 압축 풀기

```
[root@master 다운로드]# tar xvfz apache-hive-1.0.1-bin.tar.gz
```

- maria DB 설정 세팅 및 root 사용자 생성

```
[root@master 다운로드]# systemctl restart mysql
[root@master 다운로드]# systemctl status mysql
mysql.service - LSB: start and stop MySQL
   Loaded: loaded (/etc/rc.d/init.d/mysql)
   Active: active (running) since 금 2018-03-23 12:58:16 KST; 5s ago
  Process: 7614 ExecStart=/etc/rc.d/init.d/mysql start (code=exited, status=0/SU
CCESS)
   CGroup: /system.slice/mysql.service
           ├7619 /bin/sh /usr/bin/mysqld_safe --datadir=/var/lib/mysql --pid...
           └7693 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --...

 3월 23 12:58:15 master systemd[1]: Starting LSB: start and stop MySQL...
 3월 23 12:58:16 master mysql[7614]: Starting MySQL. SUCCESS!
 3월 23 12:58:16 master systemd[1]: Started LSB: start and stop MySQL.
```

```
[root@master 다운로드]# chkconfig mysql on
[root@master 다운로드]# mysqladmin -u root password '111111'
[root@master 다운로드]# mysql -u root -p mysql
Enter password:
```

- 로컬호스트에서 접속하는 hive에게 권한 부여

```
MariaDB [mysql]> grant all privileges on *.* to hive@'localhost' identified by '
111111';
Query OK, 0 rows affected (0.00 sec)
```

- hive_db 생성 및 hive에게 권한 부여

```
MariaDB [mysql]> create database hive_db;
Query OK, 1 row affected (0.00 sec)

MariaDB [mysql]> grant all privileges on hive_db.* to 'hive'@'%' identified by '
111111' with grant option;
Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> grant all privileges on hive_db.* to 'hive'@'localhost' identif
ied by '111111' with grant option;
Query OK, 0 rows affected (0.01 sec)

MariaDB [mysql]> flush privileges;
Query OK, 0 rows affected (0.01 sec)

MariaDB [mysql]> commit;
Query OK, 0 rows affected (0.00 sec)
```

- /usr/local/hive/conf 에 hive-site.xml 생성 후, 텍스트 저장(hive)는 다운로드한 apache-hive01.0.1-bin 파일

```
[root@master 다운로드]# cp -r apache-hive-1.0.1-bin/ /usr/local/hive
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
    <property>
        <name>hive.metastore.local</name>
        <value>true</value>
        <description>controls whether to connect to remove metastore server or o
pen a new metastore server in Hive Client JVM</description>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionURL</name>
        <value>jdbc:mariadb://localhost:3306/hive_db?createDatabaseIfNotExist=tr
ue</value>
        <description>JDBC connect string for a JDBC metastore</description>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionDriverName</name>
        <value>org.mariadb.jdbc.Driver</value>
        <description>Driver class name for a JDBC metastore</description>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionUserName</name>
        <value>hive</value>
        <description>username to use against metastore database</description>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionPassword</name>
        <value>111111</value>
        <description>password to use against metastore database</description>
    </property>
</configuration>
```

- mariadb-java-client 파일을 hive/lib으로 옮기기

```
[root@master 다운로드]# cp mariadb-java-client-1.3.5.jar /usr/local/hive/lib
```

- hadoop와 hive 연결을 위한 작업

```
[root@master lib]# hadoop fs -mkdir /tmp/hive
[root@master lib]# hadoop fs -mkdir /user/hive/warehouse
[root@master lib]# hadoop fs -chmod 777 /tmp
[root@master lib]# hadoop fs -chmod 777 /tmp/hive
[root@master lib]# hadoop fs -chmod 777 /user/hive
[root@master lib]# hadoop fs -chmod 777 /user/hive/warehouse
[root@master lib]#
```

- hive 실행

```
[root@master conf]# hive
18/03/23 13:42:08 WARN conf.HiveConf: DEPRECATED: Configuration property hive.me
tastore.local no longer has any effect. Make sure to provide a valid value for h
ive.metastore.uris if you are connecting to a remote metastore.
18/03/23 13:42:08 WARN conf.HiveConf: HiveConf of name hive.metastore.local does
 not exist

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-com
mon-1.0.1.jar!/hive-log4j.properties
hive>
```

6. Java Application 연동 테스트

- 테이블 생성

```
hive> CREATE TABLE airline_delay(
Year INT,
MONTH INT,
DayofMonth INT,
DayofWeek INT,
DepTime INT,
CRSDepTime INT,
ArrTime INT,
CRSArrTime INT,
UniqueCarrier STRING,
FlightNum INT,
TailNum STRING,
ActualElapsedTime INT,
CRSElapsedTime INT,
AirTime INT,
ArrDelay INT,
DepDelay INT,
Origin STRING,
Dest STRING,
Distance INT
```

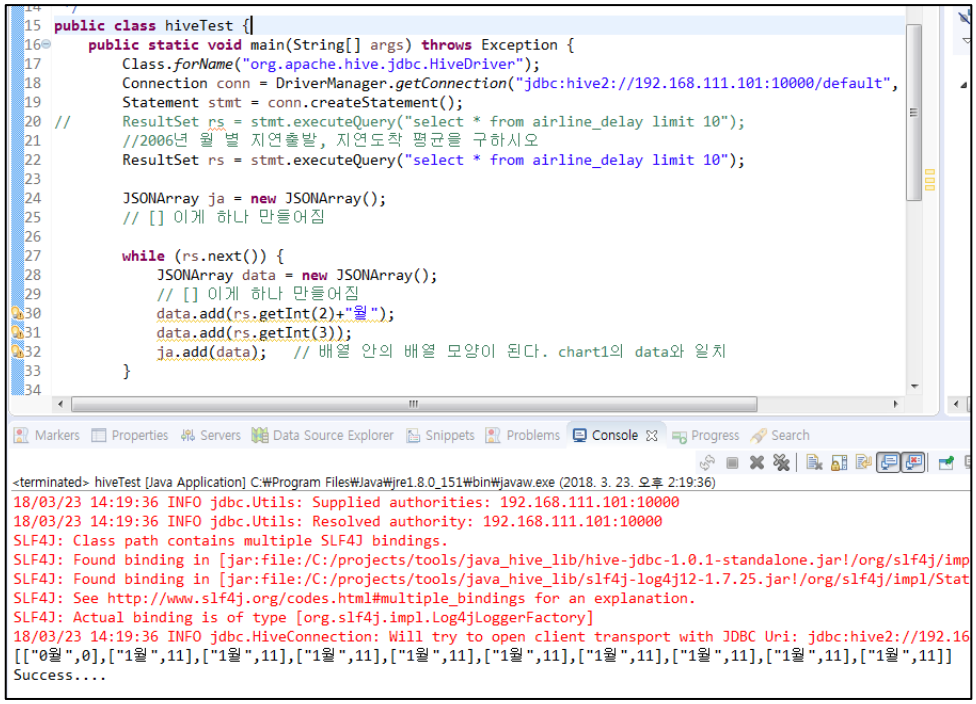
- data load

```
hive> LOAD DATA LOCAL INPATH '/root/airline/2008.csv'
    > OVERWRITE INTO TABLE airline_delay
    > PARTITION (delayYear='2008');
Loading data to table default.airline_delay partition (delayyear=2008)
Partition default.airline_delay{delayyear=2008} stats: [numFiles=1, numRows=0, t
otalSize=689413344, rawDataSize=0]
OK
Time taken: 13.296 seconds
hive> LOAD DATA LOCAL INPATH '/root/airline/2007.csv'
    > OVERWRITE INTO TABLE airline_delay
    > PARTITION (delayYear='2007');
Loading data to table default.airline_delay partition (delayyear=2007)
Partition default.airline_delay{delayyear=2007} stats: [numFiles=1, numRows=0, t
otalSize=702878193, rawDataSize=0]
OK
Time taken: 25.089 seconds
hive> LOAD DATA LOCAL INPATH '/root/airline/2006.csv'
    > OVERWRITE INTO TABLE airline_delay
    > PARTITION (delayYear='2006');
Loading data to table default.airline_delay partition (delayyear=2006)
```

- service 실행

```
[root@master conf]# hive --service hiveserver2
```

- 연동 화면



```
public class hiveTest {
    public static void main(String[] args) throws Exception {
        Class.forName("org.apache.hive.jdbc.HiveDriver");
        Connection conn = DriverManager.getConnection("jdbc:hive2://192.168.111.101:10000/default",
        Statement stmt = conn.createStatement();
//      ResultSet rs = stmt.executeQuery("select * from airline_delay limit 10");
        //2006년 월 별 지연출발, 지연도착 평균을 구하시오
        ResultSet rs = stmt.executeQuery("select * from airline_delay limit 10");

        JSONArray ja = new JSONArray();
        // [] 이게 하나 만들어짐

        while (rs.next()) {
            JSONArray data = new JSONArray();
            // [] 이게 하나 만들어짐
            data.add(rs.getInt(2)+"월");
            data.add(rs.getInt(3));
            ja.add(data);   // 배열 안의 배열 모양이 된다. chart1의 data와 일치
        }
```

Markers Properties Servers Data Source Explorer Snippets Problems Console Progress Search

```
<terminated> hiveTest [Java Application] C:₩Program Files₩Java₩jre1.8.0_151₩bin₩javaw.exe (2018. 3. 23. 오후 2:19:36)
18/03/23 14:19:36 INFO jdbc.Utils: Supplied authorities: 192.168.111.101:10000
18/03/23 14:19:36 INFO jdbc.Utils: Resolved authority: 192.168.111.101:10000
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/projects/tools/java_hive_lib/hive-jdbc-1.0.1-standalone.jar!/org/slf4j/imp
SLF4J: Found binding in [jar:file:/C:/projects/tools/java_hive_lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/Stat
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
18/03/23 14:19:36 INFO jdbc.HiveConnection: Will try to open client transport with JDBC Uri: jdbc:hive2://192.16
[["0월",0],["1월",11],["1월",11],["1월",11],["1월",11],["1월",11],["1월",11],["1월",11],["1월",11],["1월",11]]
Success....
```