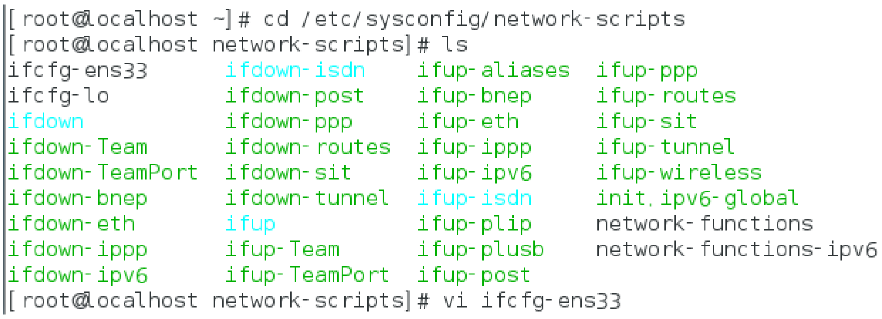
0. 편의를 위한 환경 설정 : (

전모드 해제) root > 설정 > 전원 > 빈 화면: 하지 않기 / 소프트웨어 업데이트 끄기(114p)

1. IP 주소 설정: 킬 때마다 변경되는 걸 고정시켜버리기 위함. (VMWare의 IP를 설정하고 CentOS 의 IP를 고정 시킨 것)

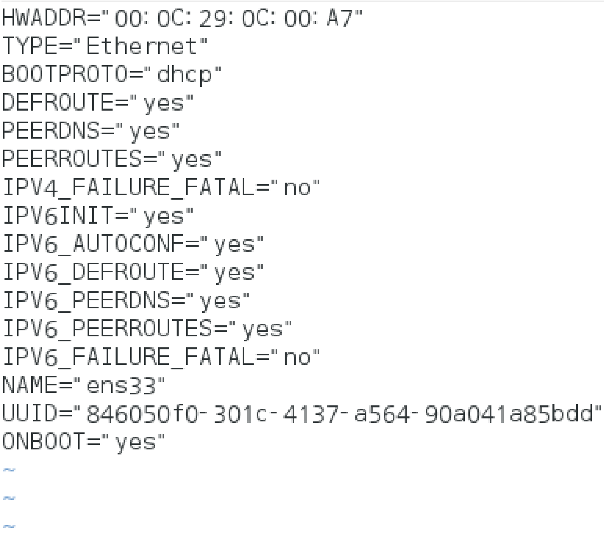
경로는 아래.



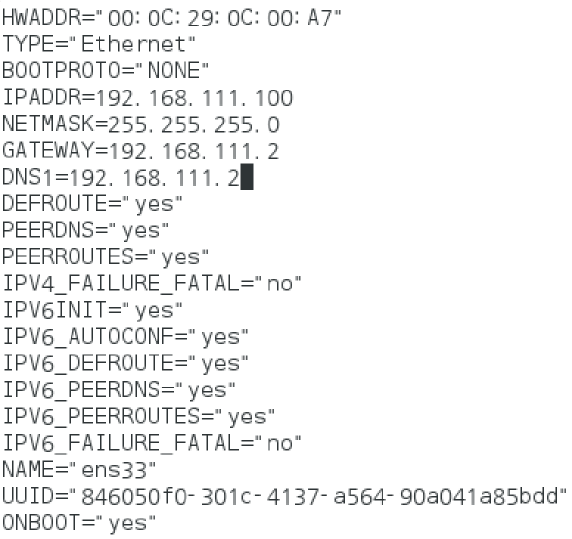
```
[root@localhost ~]# cd /etc/sysconfig/network-scripts
[root@localhost network-scripts]# ls
ifcfg-ens33       ifdown-isdn     ifup-aliases  ifup-ppp
ifcfg-lo          ifdown-post     ifup-bnep     ifup-routes
ifdown            ifdown-ppp      ifup-eth      ifup-sit
ifdown-Team       ifdown-routes   ifup-ippp     ifup-tunnel
ifdown-TeamPort   ifdown-sit      ifup-ipv6     ifup-wireless
ifdown-bnep       ifdown-tunnel   ifup-isdn     init.ipv6-global
ifdown-eth        ifup            ifup-plip     network-functions
ifdown-ippp       ifup-Team       ifup-plusb    network-functions-ipv6
ifdown-ipv6       ifup-TeamPort   ifup-post
[root@localhost network-scripts]# vi ifcfg-ens33
```

<하둡 설치 파일 참고>

원래 파일을 오른쪽과 같이 수정 (리눅스 교재 67p)..

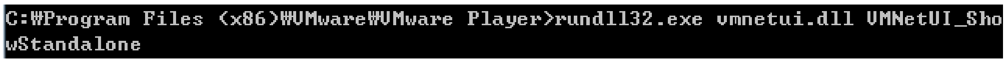


```
HWADDR="00:0C:29:0C:00:A7"
TYPE="Ethernet"
BOOTPROTO="dhcp"
DEFROUTE="yes"
PEERDNS="yes"
PEERROUTES="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_PEERDNS="yes"
IPV6_PEERROUTES="yes"
IPV6_FAILURE_FATAL="no"
NAME="ens33"
UUID="846050f0-301c-4137-a564-90a041a85bdd"
ONBOOT="yes"
~
~
~
```



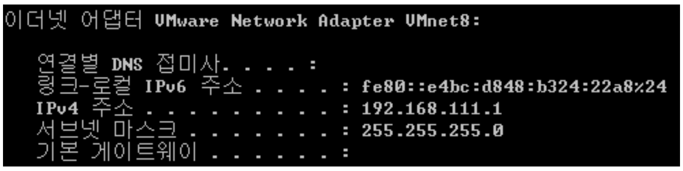
```
HWADDR="00:0C:29:0C:00:A7"
TYPE="Ethernet"
BOOTPROTO="NONE"
IPADDR=192.168.111.100
NETMASK=255.255.255.0
GATEWAY=192.168.111.2
DNS1=192.168.111.2
DEFROUTE="yes"
PEERDNS="yes"
PEERROUTES="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_PEERDNS="yes"
IPV6_PEERROUTES="yes"
IPV6_FAILURE_FATAL="no"
NAME="ens33"
UUID="846050f0-301c-4137-a564-90a041a85bdd"
ONBOOT="yes"
```

이후 책 68~69p 진행

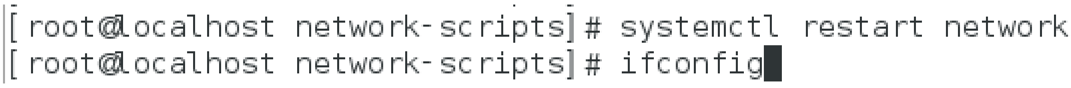


```
C:₩Program Files (x86)₩VMware₩VMware Player>rundll32.exe vmnetui.dll VMNetUI_Sho
wStandalone
```

ipconfig 입력하면 VMnet8이 아래와 같다.



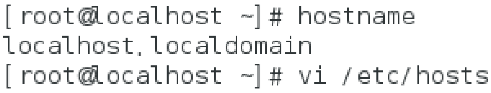
```
이더넷 어댑터 VMware Network Adapter VMnet8:

   연결별 DNS 접미사. . . . :
   링크-로컬 IPv6 주소 . . . . : fe80::e4bc:d848:b324:22a8%24
   IPv4 주소 . . . . . . . . . : 192.168.111.1
   서브넷 마스크 . . . . . . . : 255.255.255.0
   기본 게이트웨이 . . . . . . :
```



```
[root@localhost network-scripts]# systemctl restart network
[root@localhost network-scripts]# ifconfig
```

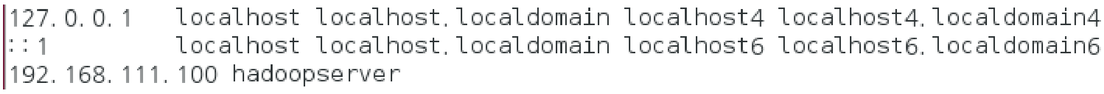
파이어폭스 열어서 인터넷 연결 잘되면 정상적인 것.

2. 하둡 서버 설정

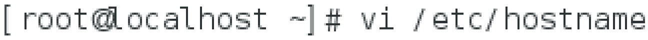


```
[root@localhost ~]# hostname
localhost.localdomain
[root@localhost ~]# vi /etc/hosts
```

vi /etc/hosts : 마지막 한 줄 추가



```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.111.100 hadoopserver
```

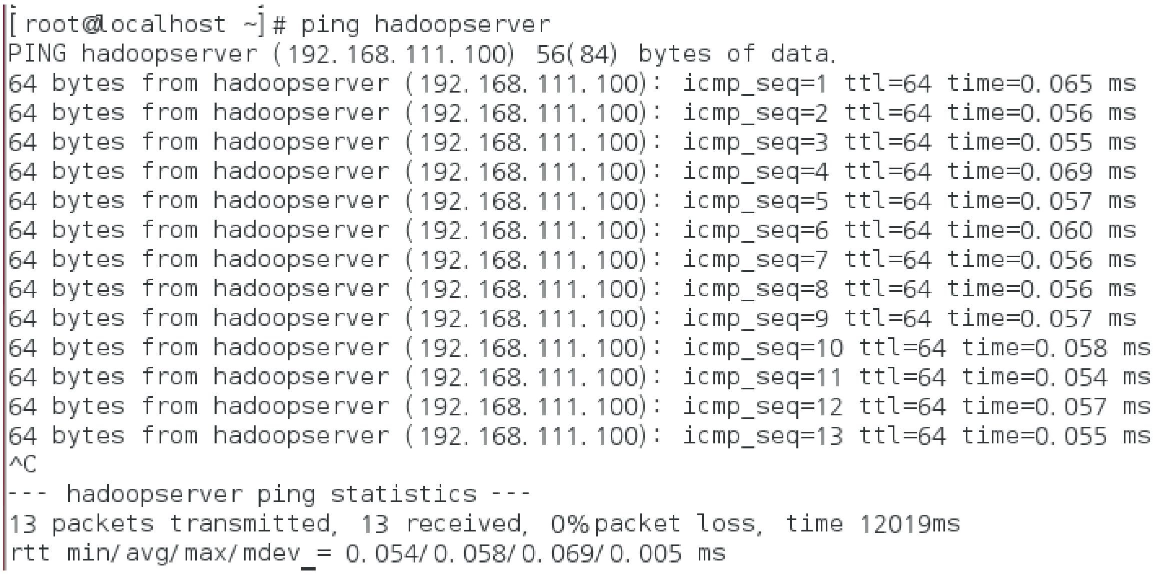


```
[root@localhost ~]# vi /etc/hostname
```

vi /etc/hostname : 변경

hadoopserver로 변경

잘 되는지 확인 하려면 ping hadoopserver 해보기(중지는 Ctrl + C)



```
[root@localhost ~]# ping hadoopserver
PING hadoopserver (192.168.111.100) 56(84) bytes of data.
64 bytes from hadoopserver (192.168.111.100): icmp_seq=1 ttl=64 time=0.065 ms
64 bytes from hadoopserver (192.168.111.100): icmp_seq=2 ttl=64 time=0.056 ms
64 bytes from hadoopserver (192.168.111.100): icmp_seq=3 ttl=64 time=0.055 ms
64 bytes from hadoopserver (192.168.111.100): icmp_seq=4 ttl=64 time=0.069 ms
64 bytes from hadoopserver (192.168.111.100): icmp_seq=5 ttl=64 time=0.057 ms
64 bytes from hadoopserver (192.168.111.100): icmp_seq=6 ttl=64 time=0.060 ms
64 bytes from hadoopserver (192.168.111.100): icmp_seq=7 ttl=64 time=0.056 ms
64 bytes from hadoopserver (192.168.111.100): icmp_seq=8 ttl=64 time=0.056 ms
64 bytes from hadoopserver (192.168.111.100): icmp_seq=9 ttl=64 time=0.057 ms
64 bytes from hadoopserver (192.168.111.100): icmp_seq=10 ttl=64 time=0.058 ms
64 bytes from hadoopserver (192.168.111.100): icmp_seq=11 ttl=64 time=0.054 ms
64 bytes from hadoopserver (192.168.111.100): icmp_seq=12 ttl=64 time=0.057 ms
64 bytes from hadoopserver (192.168.111.100): icmp_seq=13 ttl=64 time=0.055 ms
^C
--- hadoopserver ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12019ms
rtt min/avg/max/mdev = 0.054/0.058/0.069/0.005 ms
```

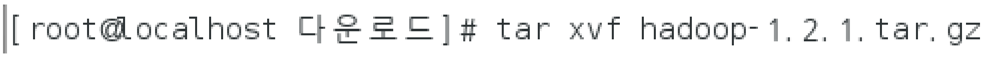
3 자바 다운로드 : 오라클 downloads > SE > JDK > Linux 64bit 다운로드

4. 하둡 다운로드 : wget http://apache.mirror.cdnetworks.com/hadoop/core/hadoop-1.2.1/hadoop-1.2.1.tar.gz

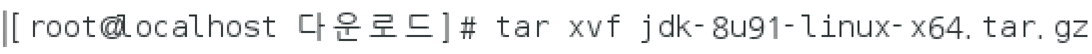
5. 아파치 톰캣 설치 : tomcat.apache.org에서 다운로드 Tomcat8.0 가서 8.0.36에 tar.gz 받음.

6. 이클립스 설치 : eclipse.org 가서 다운로드 > download packages 에서 Java EE 64bit 다운

7. 압축 풀기

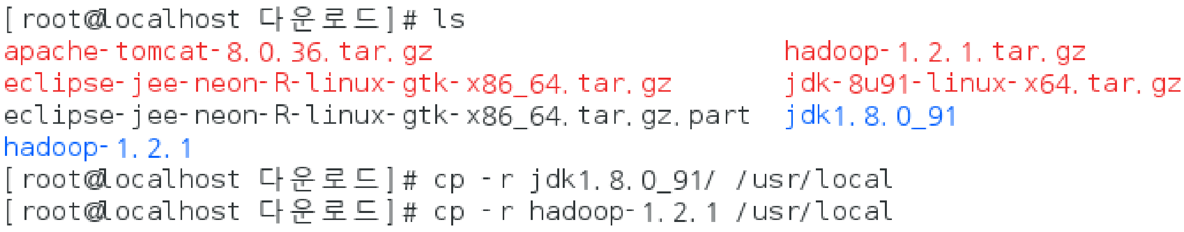


```
[root@localhost 다운로드]# tar xvf hadoop-1.2.1.tar.gz
```



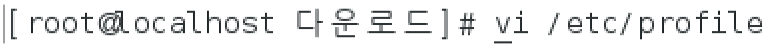
```
[root@localhost 다운로드]# tar xvf jdk-8u91-linux-x64.tar.gz
```

8. 폴더 복사



```
[root@localhost 다운로드]# ls
apache-tomcat-8.0.36.tar.gz                          hadoop-1.2.1.tar.gz
eclipse-jee-neon-R-linux-gtk-x86_64.tar.gz           jdk-8u91-linux-x64.tar.gz
eclipse-jee-neon-R-linux-gtk-x86_64.tar.gz.part      jdk1.8.0_91
hadoop-1.2.1
[root@localhost 다운로드]# cp -r jdk1.8.0_91/ /usr/local
[root@localhost 다운로드]# cp -r hadoop-1.2.1 /usr/local
```

9. 환경설정 파일 초기화



```
[root@localhost 다운로드]# vi /etc/profile
```

52라인에 아래와 같이 추가 (:set nu 로 줄 넘버 띄움)

하둡2.9.0버전

JAVA_HOME=/usr/local/jdk1.8

HADOOP_HOME=/usr/local/hadoop-2.9.0

HIVE_HOME=/usr/local/hive

CLASSPATH=/usr/local/jdk1.8/lib

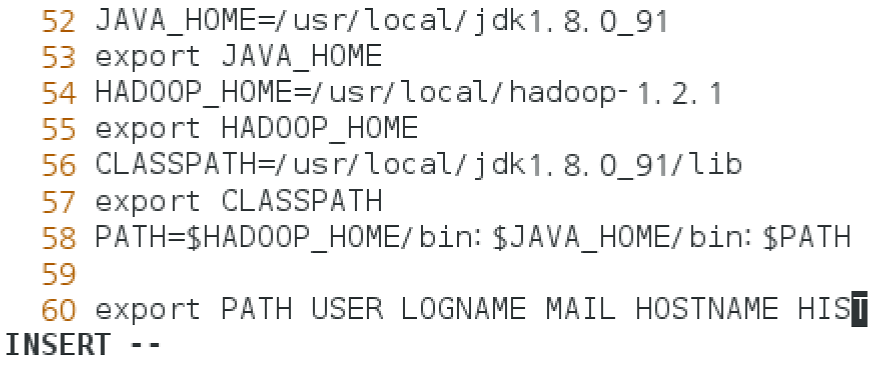
export JAVA_HOME

export HIVE_HOME

export HADOOP_HOME

export CLASSPATH

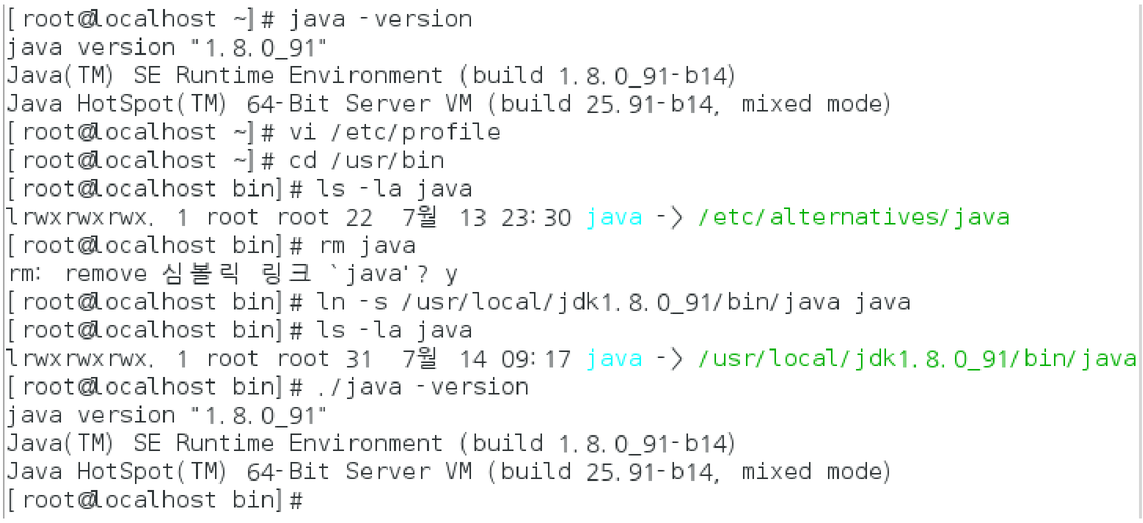
PATH=$HADOOP_HOME/bin:$JAVA_HOME/bin:$HADOOP_HOME/sbin:$HIVE_HOME/bin:$PATH



```
  52 JAVA_HOME=/usr/local/jdk1.8.0_91
  53 export JAVA_HOME
  54 HADOOP_HOME=/usr/local/hadoop-1.2.1
  55 export HADOOP_HOME
  56 CLASSPATH=/usr/local/jdk1.8.0_91/lib
  57 export CLASSPATH
  58 PATH=$HADOOP_HOME/bin:$JAVA_HOME/bin:$PATH
  59
  60 export PATH USER LOGNAME MAIL HOSTNAME HIST
INSERT --
```

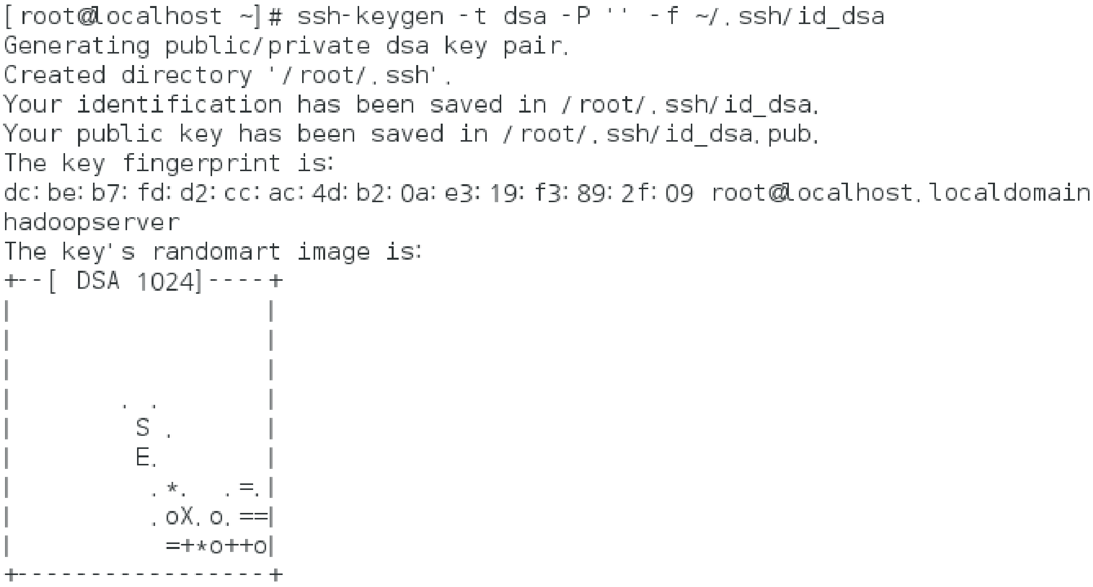
ln -s /usr/local/jdk1.8.0_161/bin/java java

java –version을 실행했을 때 1.8이 안 나오면 profile을 확인해줘야 한다. 1.8버전임을 확인했으면 심볼릭 링크를 재설정한다.

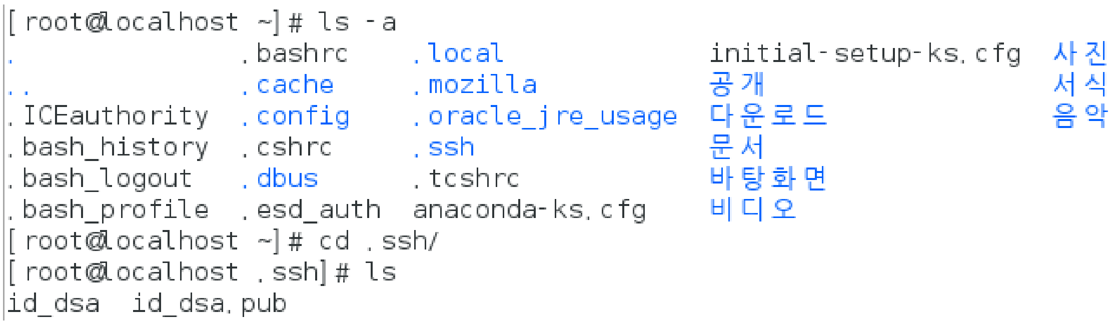


```
[root@localhost ~]# java -version
java version "1.8.0_91"
Java(TM) SE Runtime Environment (build 1.8.0_91-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.91-b14, mixed mode)
[root@localhost ~]# vi /etc/profile
[root@localhost ~]# cd /usr/bin
[root@localhost bin]# ls -la java
lrwxrwxrwx. 1 root root 22  7월 13 23:30 java -> /etc/alternatives/java
[root@localhost bin]# rm java
rm: remove 심볼릭 링크 `java'? y
[root@localhost bin]# ln -s /usr/local/jdk1.8.0_91/bin/java java
[root@localhost bin]# ls -la java
lrwxrwxrwx. 1 root root 31  7월 14 09:17 java -> /usr/local/jdk1.8.0_91/bin/java
[root@localhost bin]# ./java -version
java version "1.8.0_91"
Java(TM) SE Runtime Environment (build 1.8.0_91-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.91-b14, mixed mode)
[root@localhost bin]#
```

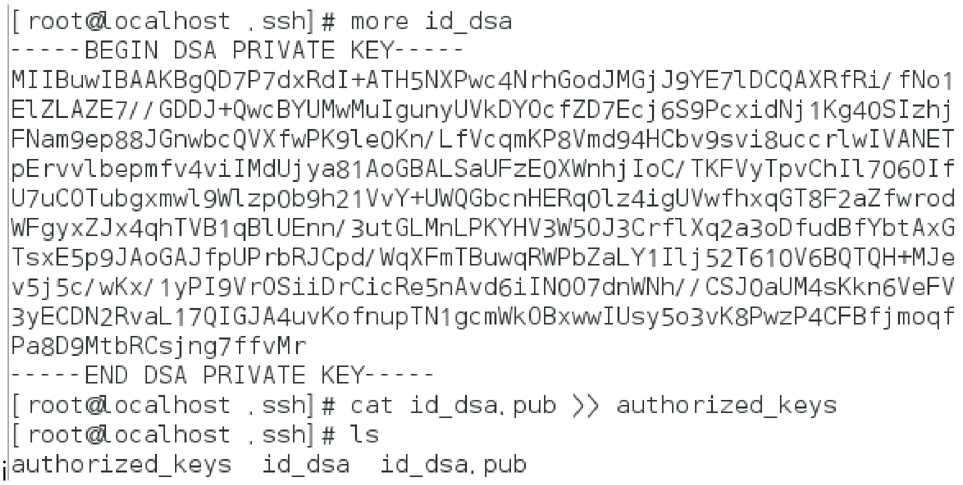
10. SSH



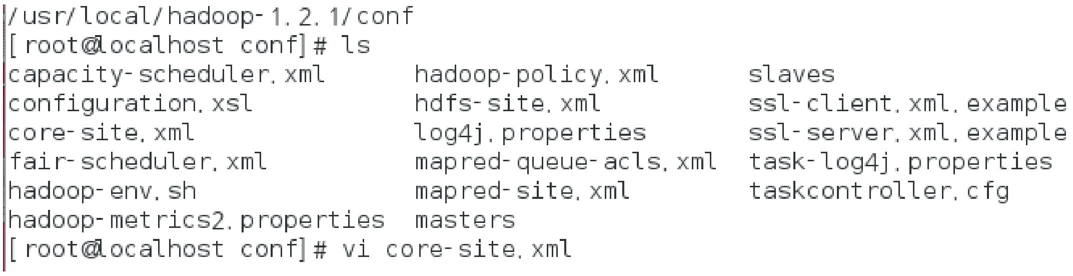
```
[root@localhost ~]# ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
Generating public/private dsa key pair.
Created directory '/root/.ssh'.
Your identification has been saved in /root/.ssh/id_dsa.
Your public key has been saved in /root/.ssh/id_dsa.pub.
The key fingerprint is:
dc:be:b7:fd:d2:cc:ac:4d:b2:0a:e3:19:f3:89:2f:09 root@localhost.localdomain
hadoopserver
The key's randomart image is:
+--[ DSA 1024]----+
|                 |
|                 |
|                 |
|       . .       |
|        S .      |
|        E.       |
|         .*.  .=.|
|         .oX.o.==|
|          =+*o++o|
+-----------------+
```



```
[root@localhost ~]# ls -a
.               .bashrc     .local              initial-setup-ks.cfg  사진
..              .cache      .mozilla            공개                  서식
.ICEauthority   .config     .oracle_jre_usage   다운로드              음악
.bash_history   .cshrc      .ssh                문서
.bash_logout    .dbus       .tcshrc             바탕화면
.bash_profile   .esd_auth   anaconda-ks.cfg     비디오
[root@localhost ~]# cd .ssh/
[root@localhost .ssh]# ls
id_dsa  id_dsa.pub
```



```
[root@localhost .ssh]# more id_dsa
-----BEGIN DSA PRIVATE KEY-----
MIIBuwIBAAKBgQD7P7dxRdI+ATH5NXPwc4NrhGodJMGjJ9YE7lDCQAXRfRi/fNo1
ElZLAZE7//GDDJ+QwcBYUMwMuIgunyUVkDYOcfZD7Ecj6S9PcxidNj1Kg40SIzhj
FNam9ep88JGnwbcQVXfwPK9le0Kn/LfVcqmKP8Vmd94HCbv9svi8uccrlwIVANET
pErvvlbepmfv4viIMdUjya81AoGBALSaUFzE0XWnhjIoC/TKFVyTpvChIl7O6OIf
U7uCOTubgxmwl9Wlzp0b9h21VvY+UWQGbcnHERq0lz4igUVwfhxqGT8F2aZfwrod
WFgyxZJx4qhTVB1qBlUEnn/3utGLMnLPKYHV3W5OJ3CrflXq2a3oDfudBfYbtAxG
TsxE5p9JAoGAJfpUPrbRJCpd/WqXFmTBuwqRWPbZaLY1Ilj52T61OV6BQTQH+MJe
v5j5c/wKx/1yPI9Vr0SiiDrCicRe5nAvd6iINOO7dnWNh//CSJ0aUM4sKkn6VeFV
3yECDN2RvaL17QIGJA4uvKofnupTN1gcmWkOBxwwIUsy5o3vK8PwzP4CFBfjmoqf
Pa8D9MtbRCsjng7ffvMr
-----END DSA PRIVATE KEY-----
[root@localhost .ssh]# cat id_dsa.pub >> authorized_keys
[root@localhost .ssh]# ls
authorized_keys  id_dsa  id_dsa.pub
```



```
/usr/local/hadoop-1.2.1/conf
[root@localhost conf]# ls
capacity-scheduler.xml      hadoop-policy.xml      slaves
configuration.xsl           hdfs-site.xml          ssl-client.xml.example
core-site.xml               log4j.properties       ssl-server.xml.example
fair-scheduler.xml          mapred-queue-acls.xml  task-log4j.properties
hadoop-env.sh               mapred-site.xml        taskcontroller.cfg
hadoop-metrics2.properties  masters
[root@localhost conf]# vi core-site.xml
```

아래와 같이 추가

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9000</value>

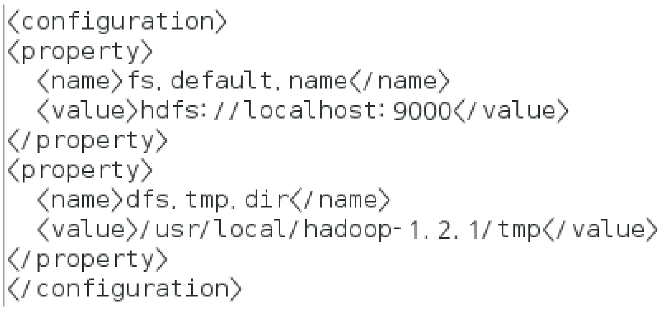
</property>

<property>

<name>dfs.tmp.dir</name>

<value>/usr/local/hadoop-1.2.1/tmp</value>

</property>



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

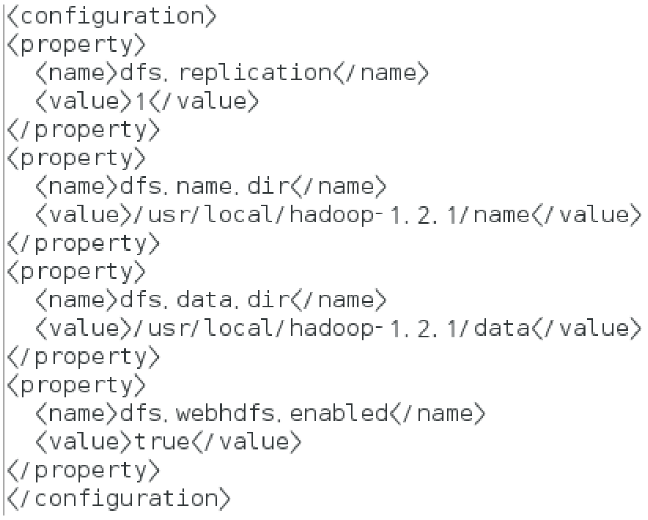
vi hdfs-site.xml 해서 아래와 같이 내용 추가

```
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.http.address</name>
<value>localhost: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>
<property>
  <name>dfs.replication</name>
  <value>1</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>
```

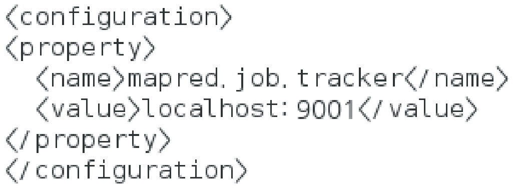
vi mapred-site.xml에 아래와 같이 추가

<property>

<name>mapred.job.tracker</name>

<value>localhost:9001</value>

</property>



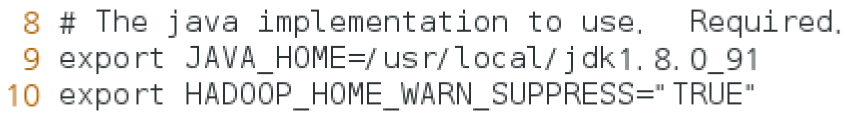
```
<configuration>
<property>
  <name>mapred.job.tracker</name>
  <value>localhost:9001</value>
</property>
</configuration>
```

하둡에 환경설정 파일

systemctl stop firewalld

systemctl disable firewalld

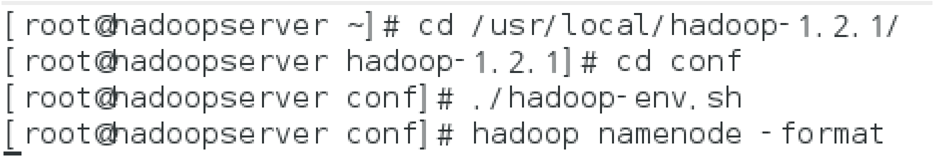
cd conf 해서 vi hadoop-env.sh 열어서 아래와 같이 추가. (# 주석 지워야함)



```
# The java implementation to use.  Required.
export JAVA_HOME=/usr/local/jdk1.8.0_91
export HADOOP_HOME_WARN_SUPPRESS="TRUE"
```

재부팅 함

Hadoop 실행시켜본 되면 포맷 안되면 안됨



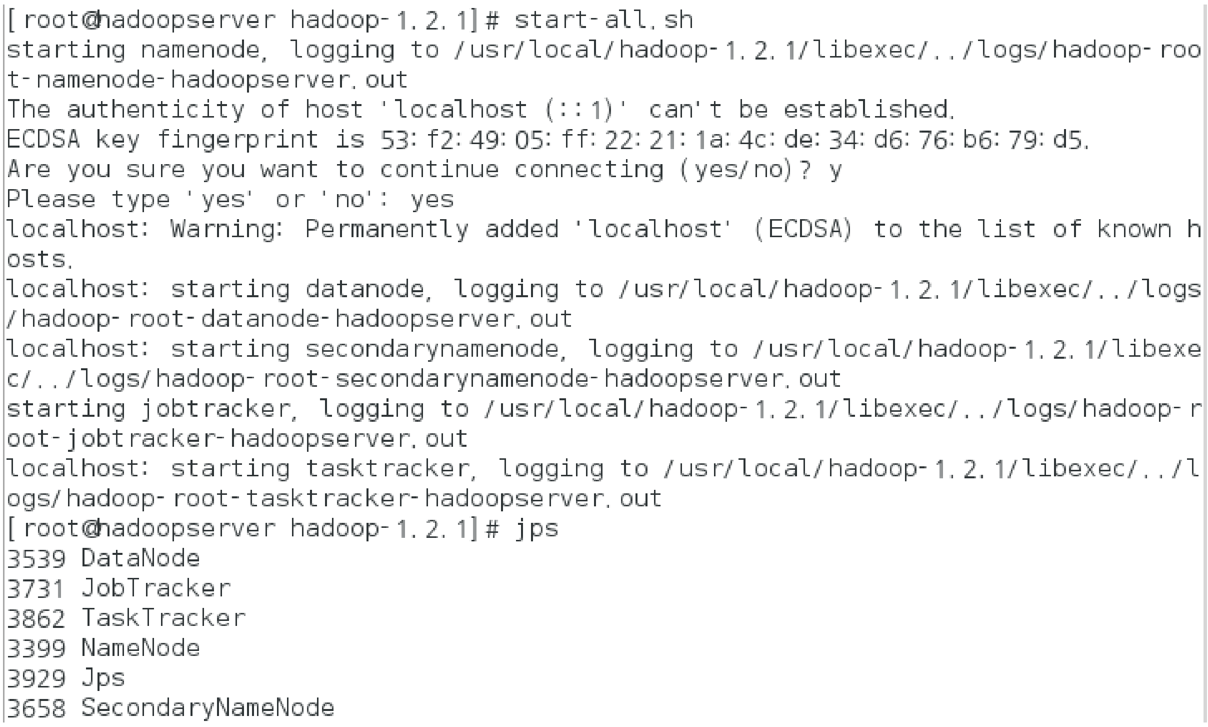
```
[root@hadoopserver ~]# cd /usr/local/hadoop-1.2.1/
[root@hadoopserver hadoop-1.2.1]# cd conf
[root@hadoopserver conf]# ./hadoop-env.sh
[root@hadoopserver conf]# hadoop namenode -format
```

아까 잘못 포맷하고 재시작해서 네임폴더가 만들어져서 제대로 안됨.

data,name,tmp 폴더를 모두 삭제 하고 다시 포멧 재 실행

start-all.sh 해서 jps로 확인했을 때 아래와 같이 6개가 뜨면 잘 되는 것.

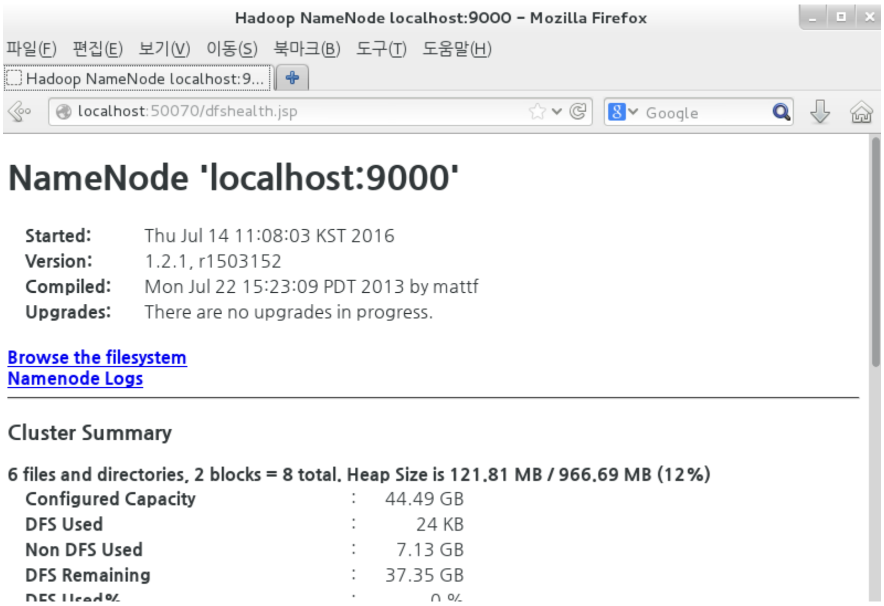
start-all.sh 는 하둡 시작 시키는 것



```
[root@hadoopserver hadoop-1.2.1]# start-all.sh
starting namenode, logging to /usr/local/hadoop-1.2.1/libexec/../logs/hadoop-roo
t-namenode-hadoopserver.out
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is 53:f2:49:05:ff:22:21:1a:4c:de:34:d6:76:b6:79:d5.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
localhost: Warning: Permanently added 'localhost' (ECDSA) to the list of known h
osts.
localhost: starting datanode, logging to /usr/local/hadoop-1.2.1/libexec/../logs
/hadoop-root-datanode-hadoopserver.out
localhost: starting secondarynamenode, logging to /usr/local/hadoop-1.2.1/libexe
c/../logs/hadoop-root-secondarynamenode-hadoopserver.out
starting jobtracker, logging to /usr/local/hadoop-1.2.1/libexec/../logs/hadoop-r
oot-jobtracker-hadoopserver.out
localhost: starting tasktracker, logging to /usr/local/hadoop-1.2.1/libexec/../l
ogs/hadoop-root-tasktracker-hadoopserver.out
[root@hadoopserver hadoop-1.2.1]# jps
3539 DataNode
3731 JobTracker
3862 TaskTracker
3399 NameNode
3929 Jps
3658 SecondaryNameNode
```

파이어 폭스에 http://localhost:50070 입력하면 요렇게 뜨지요 얘는 모니터링용.

Live node 클릭하면 현재 살아 움직이는 노드에 대해 모니터링 할 수 있다.


Hadoop NameNode localhost:9000 – Mozilla Firefox
파일(F) 편집(E) 보기(V) 이동(S) 북마크(B) 도구(T) 도움말(H)
Hadoop NameNode localhost:9...
localhost:50070/dfshealth.jsp
Google

NameNode 'localhost:9000'

Started: Thu Jul 14 11:08:03 KST 2016
Version: 1.2.1, r1503152
Compiled: Mon Jul 22 15:23:09 PDT 2013 by mattf
Upgrades: There are no upgrades in progress.

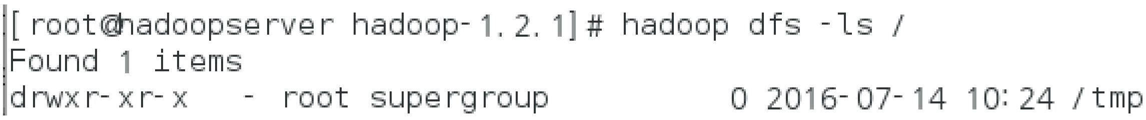
Browse the filesystem
Namenode Logs

Cluster Summary

6 files and directories, 2 blocks = 8 total. Heap Size is 121.81 MB / 966.69 MB (12%)

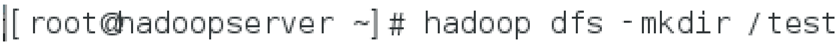
Configured Capacity	:	44.49 GB
DFS Used	:	24 KB
Non DFS Used	:	7.13 GB
DFS Remaining	:	37.35 GB
DFS Used%	:	0 %

하둡시스템에 / (루트 디렉터리)의 내용을 보자는 내용.

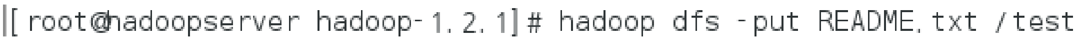


```
[root@hadoopserver hadoop-1.2.1]# hadoop dfs -ls /
Found 1 items
drwxr-xr-x   - root supergroup          0 2016-07-14 10:24 /tmp
```

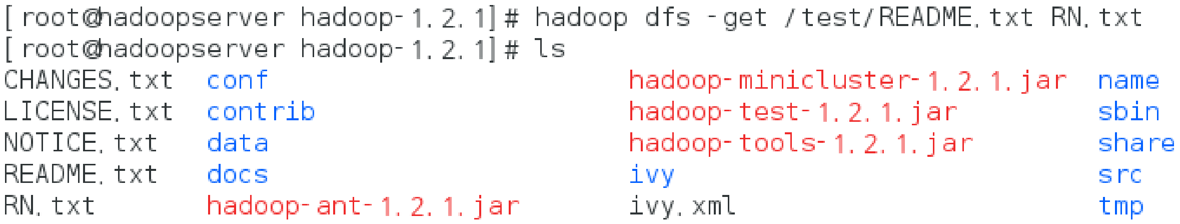
Test 라는 디렉토리를 만들어서 리드미를 거기에 올리고 가져오는 과정



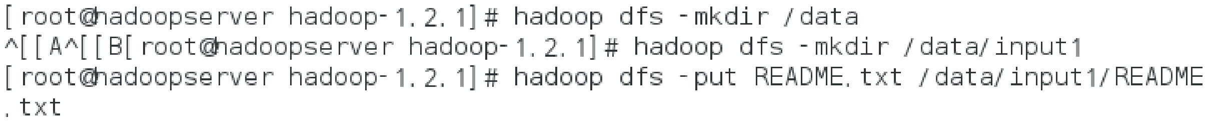
```
[root@hadoopserver ~]# hadoop dfs -mkdir /test
```



```
[root@hadoopserver hadoop-1.2.1]# hadoop dfs -put README.txt /test
```

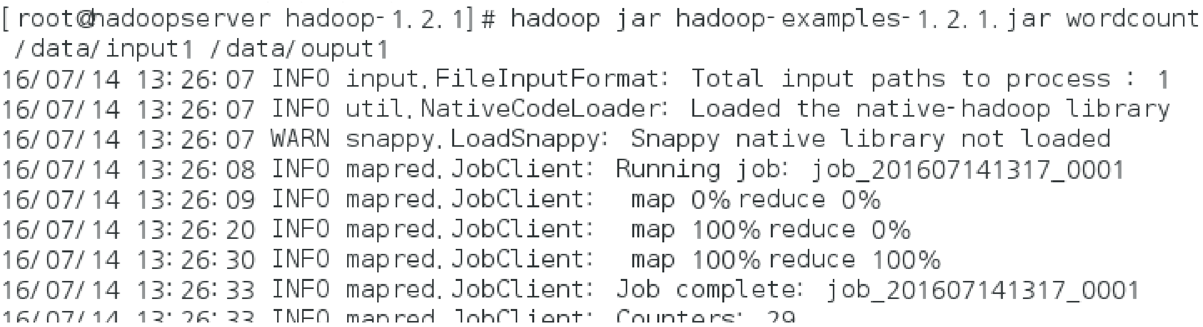


```
[root@hadoopserver hadoop-1.2.1]# hadoop dfs -get /test/README.txt RN.txt
[root@hadoopserver hadoop-1.2.1]# ls
CHANGES.txt  conf                      hadoop-minicluster-1.2.1.jar  name
LICENSE.txt  contrib                   hadoop-test-1.2.1.jar         sbin
NOTICE.txt   data                      hadoop-tools-1.2.1.jar        share
README.txt   docs                      ivy                           src
RN.txt       hadoop-ant-1.2.1.jar      ivy.xml                       tmp
```



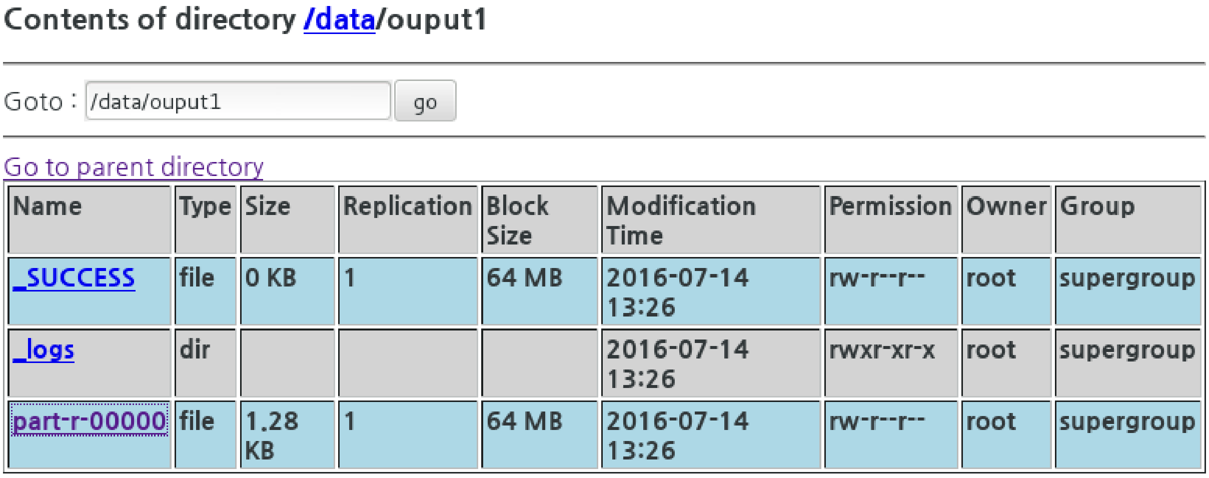
```
[root@hadoopserver hadoop-1.2.1]# hadoop dfs -mkdir /data
^[[A^[[B[root@hadoopserver hadoop-1.2.1]# hadoop dfs -mkdir /data/input1
[root@hadoopserver hadoop-1.2.1]# hadoop dfs -put README.txt /data/input1/README
.txt
```

Input1 폴더의 파일들의 워드를 세어서 output1에 저장. hadoop~.jar는 자바 어플리케이션 역할



```
[root@hadoopserver hadoop-1.2.1]# hadoop jar hadoop-examples-1.2.1.jar wordcount
 /data/input1 /data/ouput1
16/07/14 13:26:07 INFO input.FileInputFormat: Total input paths to process : 1
16/07/14 13:26:07 INFO util.NativeCodeLoader: Loaded the native-hadoop library
16/07/14 13:26:07 WARN snappy.LoadSnappy: Snappy native library not loaded
16/07/14 13:26:08 INFO mapred.JobClient: Running job: job_201607141317_0001
16/07/14 13:26:09 INFO mapred.JobClient:  map 0% reduce 0%
16/07/14 13:26:20 INFO mapred.JobClient:  map 100% reduce 0%
16/07/14 13:26:30 INFO mapred.JobClient:  map 100% reduce 100%
16/07/14 13:26:33 INFO mapred.JobClient: Job complete: job_201607141317_0001
16/07/14 13:26:33 INFO mapred.JobClient: Counters: 29
```

결과를 이렇게 확인할 수 있지 (localhost:50070 > Browse File System 에 들어가면 된다.)

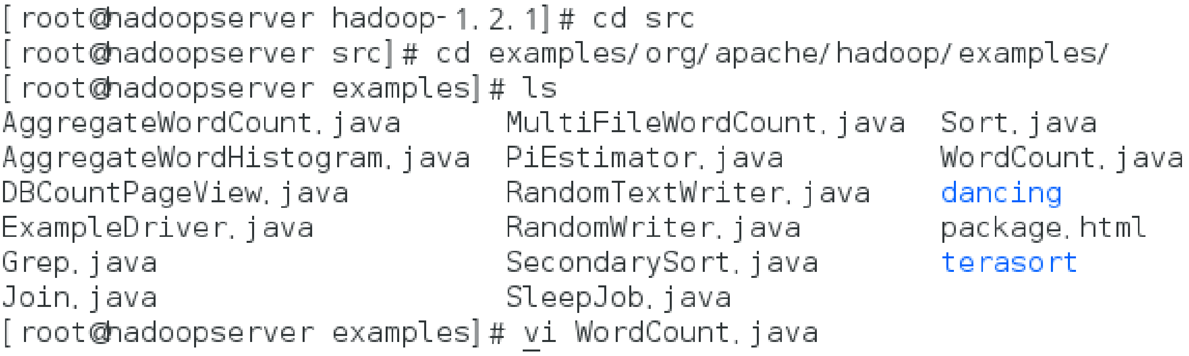


Contents of directory /data/ouput1

Goto : /data/ouput1 go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
_SUCCESS	file	0 KB	1	64 MB	2016-07-14 13:26	rw-r--r--	root	supergroup
_logs	dir				2016-07-14 13:26	rwxr-xr-x	root	supergroup
part-r-00000	file	1.28 KB	1	64 MB	2016-07-14 13:26	rw-r--r--	root	supergroup



```
[root@hadoopserver hadoop-1.2.1]# cd src
[root@hadoopserver src]# cd examples/org/apache/hadoop/examples/
[root@hadoopserver examples]# ls
AggregateWordCount.java      MultiFileWordCount.java   Sort.java
AggregateWordHistogram.java  PiEstimator.java          WordCount.java
DBCountPageView.java         RandomTextWriter.java     dancing
ExampleDriver.java           RandomWriter.java         package.html
Grep.java                    SecondarySort.java        terasort
Join.java                    SleepJob.java
[root@hadoopserver examples]# vi WordCount.java
```

HIVE 설치

리눅스 책 555~57p 따라서 마리아디비 깔기, firewall 설정까지 한다.

Cd 다운로드가서

yum -y remove mariadb-libs

yum -y localinstall Maria*

systemctl restart mysql

systemctl status mysql

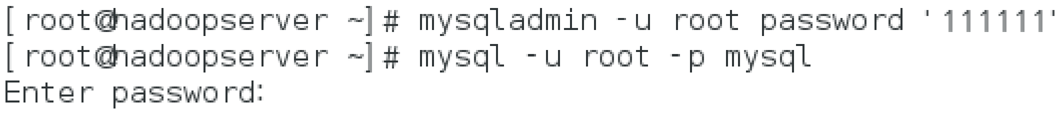
chkconfig mysql on

mysqladmin -u root password '111111'

mysql -u root -p mysql-->마리아디비mysql에 접속

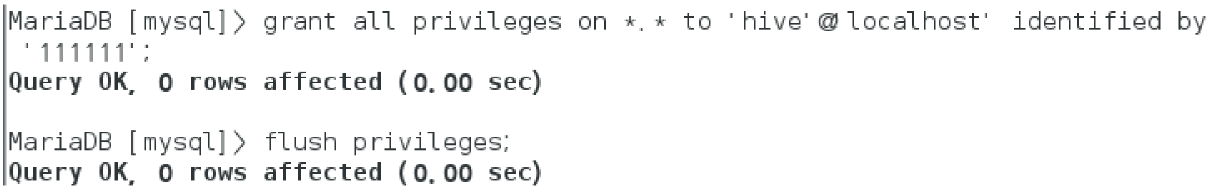
mysql -u hive -p hive_db -->마리아디비 hive_db 에 접속

(http://ftp.kaist.ac.kr/mariadb/mariadb-10.0.26/yum/centos7-amd64/rpms/)



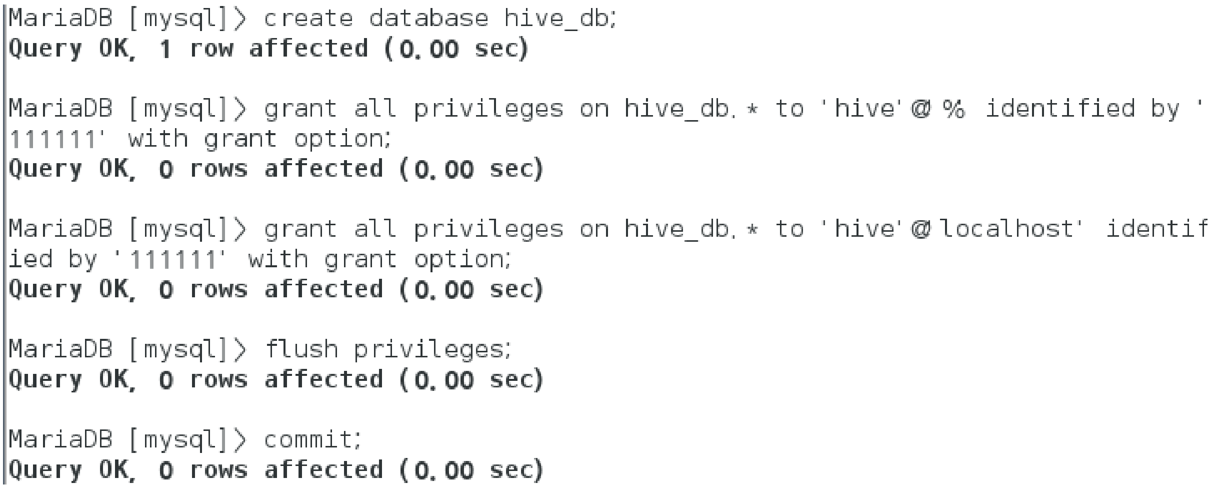
```
[root@hadoopserver ~]# mysqladmin -u root password '111111'
[root@hadoopserver ~]# mysql -u root -p mysql
Enter password:
```

로컬호스트에서 접속하는 hive에게 권한을 다 주겠다.

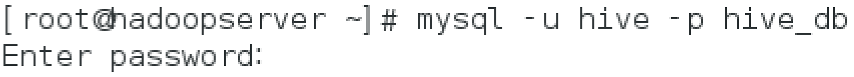


```
MariaDB [mysql]> grant all privileges on *.* to 'hive'@localhost' identified by
 '111111';
Query OK, 0 rows affected (0.00 sec)

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

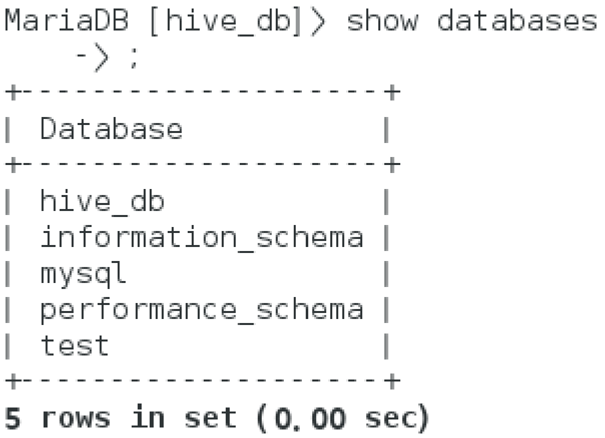
hive_db를 만들고 그 권한을 hive에게 다 줌



```
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 (0.00 sec)

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

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



```
[root@hadoopserver ~]# mysql -u hive -p hive_db
Enter password:
```

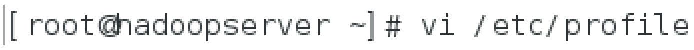


```
MariaDB [hive_db]> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| hive_db            |
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
5 rows in set (0.00 sec)
```

mysql 작업 끝

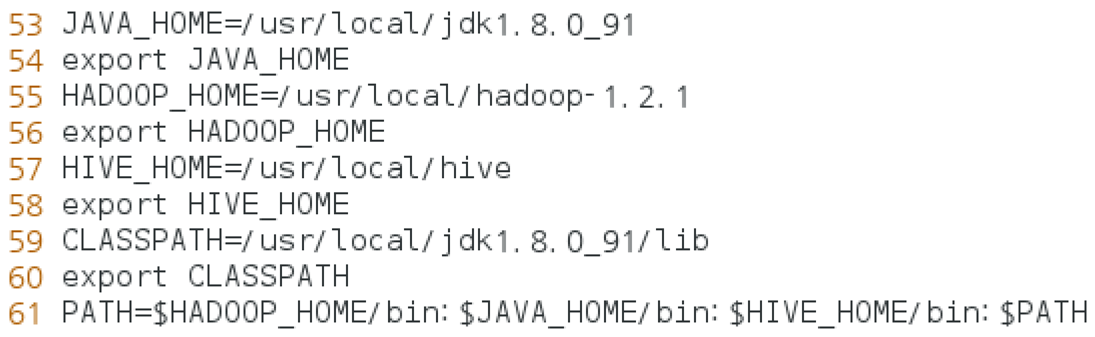
HIVE 세팅 시작합니다.

http://apache.tt.co.kr/hive/hive-1.0.1/ 에서 ~bin.tar.gz을 받는다.

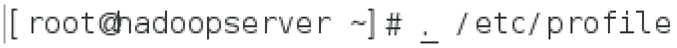


```
[root@hadoopserver ~]# vi /etc/profile
```

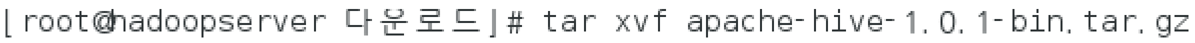
HIVE_HOME 추가 57,58라인 추가하고 61라인에 패쓰에 하이브홈을 넣어줌



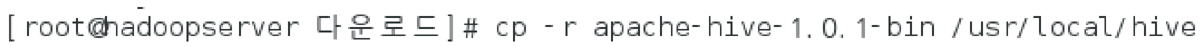
```
JAVA_HOME=/usr/local/jdk1.8.0_91
export JAVA_HOME
HADOOP_HOME=/usr/local/hadoop-1.2.1
export HADOOP_HOME
HIVE_HOME=/usr/local/hive
export HIVE_HOME
CLASSPATH=/usr/local/jdk1.8.0_91/lib
export CLASSPATH
PATH=$HADOOP_HOME/bin:$JAVA_HOME/bin:$HIVE_HOME/bin:$PATH
```



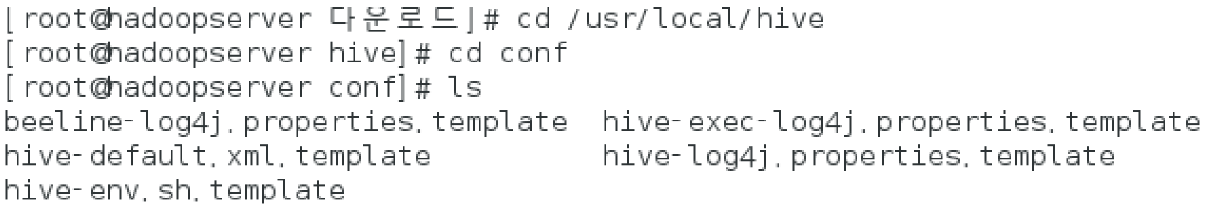
```
[root@hadoopserver ~]# . /etc/profile
```



```
[root@hadoopserver 다운로드]# tar xvf apache-hive-1.0.1-bin.tar.gz
```



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



```
[root@hadoopserver 다운로드]# cd /usr/local/hive
[root@hadoopserver hive]# cd conf
[root@hadoopserver conf]# ls
beeline-log4j.properties.template  hive-exec-log4j.properties.template
hive-default.xml.template          hive-log4j.properties.template
hive-env.sh.template
```

Hive-site.xml 만들고 메모장에 있는 내용을 넣어준다.

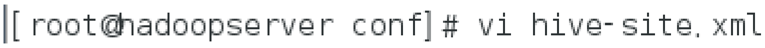
cp -r apache-hive-1.0.1-bin /usr/local/hive-1.0.1

vi /etc/profile

cd /usr/local/hive/conf

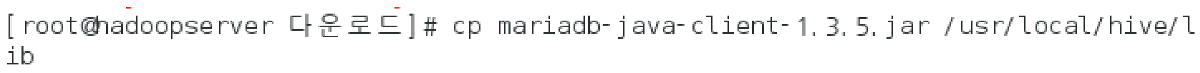
touch hive-site.xml

vi hive-site.xml



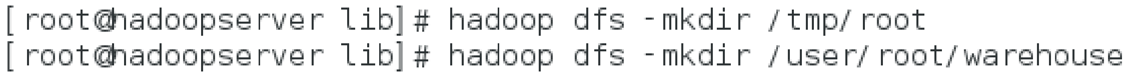
```
[root@hadoopserver conf]# vi hive-site.xml
```

MariaDB JDBC API를 받아준다. 하이브 lib 안으로 옮겨줌



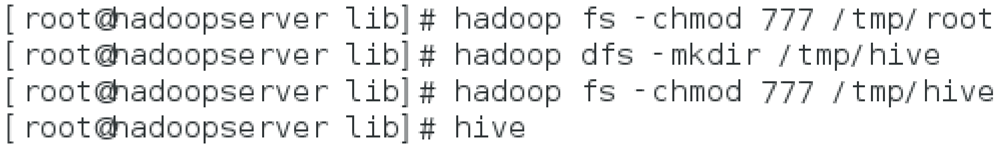
```
[root@hadoopserver 다운로드]# cp mariadb-java-client-1.3.5.jar /usr/local/hive/l
ib
```

하이브가 하둡에서 활동하는 디렉토리를 만들어줌-> 하이브와 하둡 연결



```
[root@hadoopserver lib]# hadoop dfs -mkdir /tmp/root
[root@hadoopserver lib]# hadoop dfs -mkdir /user/root/warehouse
```

Writable 할 수 있게 권한 수정



```
[root@hadoopserver lib]# hadoop fs -chmod 777 /tmp/root
[root@hadoopserver lib]# hadoop dfs -mkdir /tmp/hive
[root@hadoopserver lib]# hadoop fs -chmod 777 /tmp/hive
[root@hadoopserver lib]# hive
```

hadoop fs -mkdir /tmp/hive

hadoop fs -mkdir /user/hive/warehouse

hadoop fs -chmod 777 /tmp

hadoop fs -chmod 777 /tmp/hive

hadoop fs -chmod 777 /user/hive

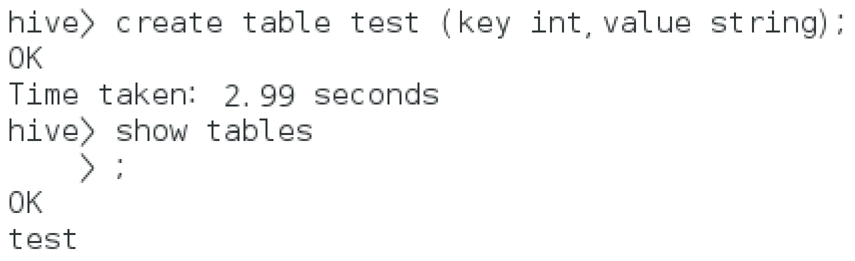
hadoop fs -chmod 777 /user/hive/warehouse

>hive 실행

-- error 후 tmp/hive 자동 생성

재부팅합시다.

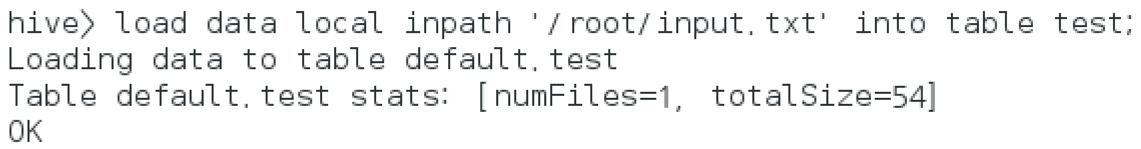
다시 start-all.sh(hadoop을 작동 시키고) 해주고 hive 접속 후 테이블을 만든다.



```
hive> create table test (key int,value string);
OK
Time taken: 2.99 seconds
hive> show tables
    > ;
OK
test
```

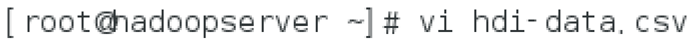
아래는 빅데이터 파일을 하이브를 통해서 하둡에 넣은 것이다.

```
LOAD DATA LOCAL INPATH '/root/airline/2008.csv'
    > OVERWRITE INTO TABLE airline_delay
    > PARTITION (delayYear='2008');
```

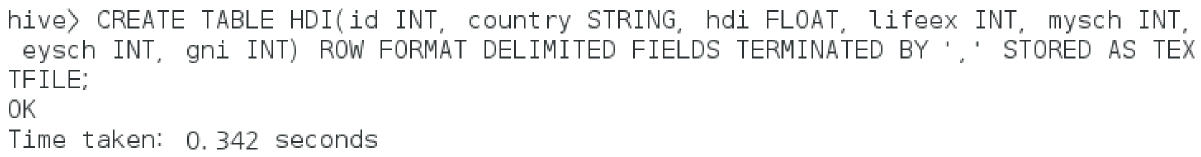


```
hive> load data local inpath '/root/input.txt' into table test;
Loading data to table default.test
Table default.test stats: [numFiles=1, totalSize=54]
OK
```

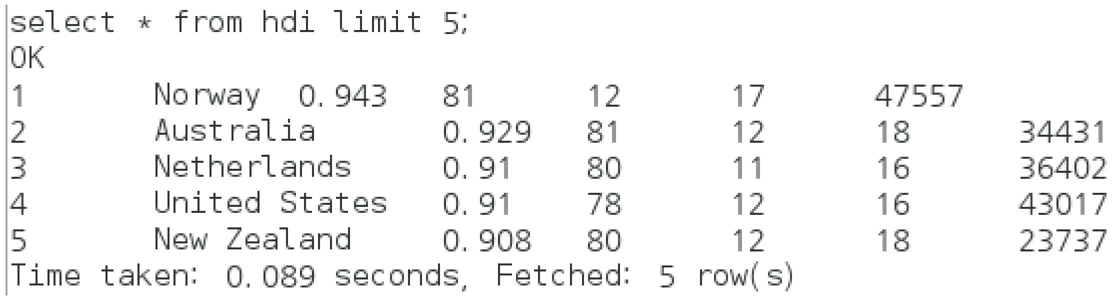
이거 말고 hdi-data.csv 파일 생성하고 테이블을 생성해서 넣으면 아래와 같이 테이블이 형성된다.



```
[root@hadoopserver ~]# vi hdi-data.csv
```

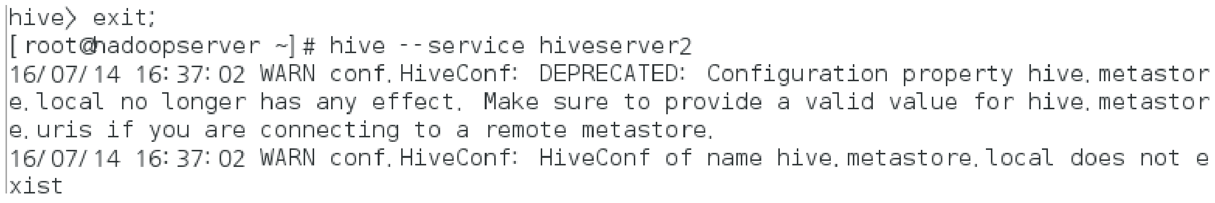


```
hive> CREATE TABLE HDI(id INT, country STRING, hdi FLOAT, lifeex INT, mysch INT,
 eysch INT, gni INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEX
TFILE;
OK
Time taken: 0.342 seconds
```



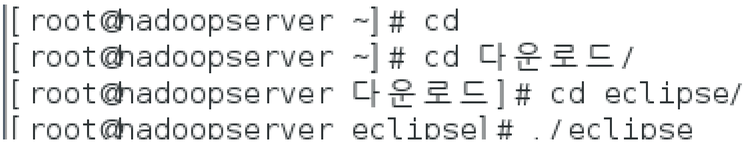
```
select * from hdi limit 5;
OK
1       Norway  0.943   81      12      17      47557
2       Australia       0.929   81      12      18      34431
3       Netherlands     0.91    80      11      16      36402
4       United States   0.91    78      12      16      43017
5       New Zealand     0.908   80      12      18      23737
Time taken: 0.089 seconds, Fetched: 5 row(s)
```

자바와 하이브가 연동될 수 있게 함.



```
hive> exit;
[root@hadoopserver ~]# hive --service hiveserver2
16/07/14 16:37:02 WARN conf.HiveConf: DEPRECATED: Configuration property hive.metastor
e.local no longer has any effect. Make sure to provide a valid value for hive.metastor
e.uris if you are connecting to a remote metastore.
16/07/14 16:37:02 WARN conf.HiveConf: HiveConf of name hive.metastore.local does not e
xist
```

이클립스 실행

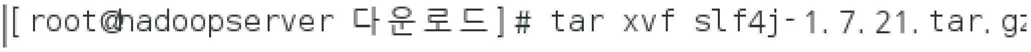


```
[root@hadoopserver ~]# cd
[root@hadoopserver ~]# cd 다운로드/
[root@hadoopserver 다운로드]# cd eclipse/
[root@hadoopserver eclipse]# ./eclipse
```

라이브러리 다운받기

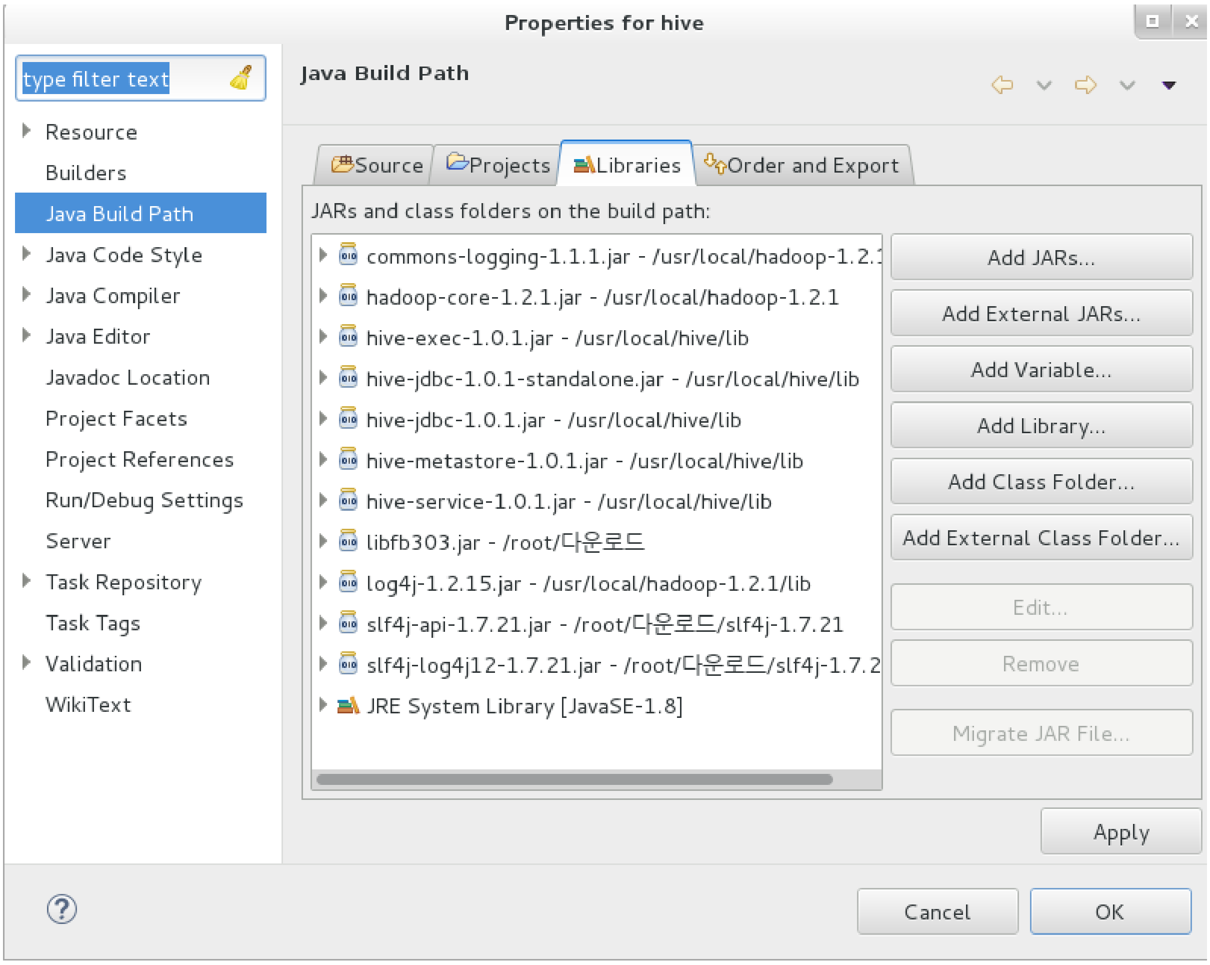
http://www.slf4j.org/download.html >> tar.gz 받고 아래처럼 압축 풀기

http://www.docjar.com/jar_detail/libfb303.jar.html >> jar 파일 받기



```
[root@hadoopserver 다운로드]# tar xvf slf4j-1.7.21.tar.g
```

이클립스 프로퍼티에 라이브러리 추가해주기



JARs and class folders on the build path:

- commons-logging-1.1.1.jar - /usr/local/hadoop-1.2.1
- hadoop-core-1.2.1.jar - /usr/local/hadoop-1.2.1
- hive-exec-1.0.1.jar - /usr/local/hive/lib
- hive-jdbc-1.0.1-standalone.jar - /usr/local/hive/lib
- hive-jdbc-1.0.1.jar - /usr/local/hive/lib
- hive-metastore-1.0.1.jar - /usr/local/hive/lib
- hive-service-1.0.1.jar - /usr/local/hive/lib
- libfb303.jar - /root/다운로드
- log4j-1.2.15.jar - /usr/local/hadoop-1.2.1/lib
- slf4j-api-1.7.21.jar - /root/다운로드/slf4j-1.7.21
- slf4j-log4j12-1.7.21.jar - /root/다운로드/slf4j-1.7.2
- JRE System Library [JavaSE-1.8]

클래스를 생성해서 메모장에서 메인에 추가해줌. 임포트도 하고 또 익셉션 쓰로우 해줌.

```
package hive;

import java.sql.*;

public class Hive {

        public static void main(String[] args) throws Exception {
                Class.forName("org.apache.hive.jdbc.HiveDriver");
                Connection conn = 
DriverManager.getConnection("jdbc:hive2://localhost:10000/default","","");
                Statement stmt = conn.createStatement();
                ResultSet rs = stmt.executeQuery("SELECT * FROM hdi");
                  while(rs.next()) {
                     System.out.println(rs.getString(1));
                  }
                conn.close();
                System.out.println("Success....");
        }

}
```

```
LOAD DATA LOCAL INPATH '/root/airline/2008.csv'
    > OVERWRITE INTO TABLE airline_delay
    > PARTITION (delayYear='2008');

SELECT * FROM airline_delay WHERE delayYear='2006'
LIMIT 20;

SELECT Year,Month,COUNT(*) FROM airline_delay
    > WHERE delayYear =2006
    > AND ArrDelay >0
    > GROUP BY Year,Month;

SELECT Year,Month ,AVG(ArrDelay), AS AVG_arr, AVG(DepDelay) AS
    > AVG_dep
    > FROM airline_delay
    > WHERE delayYear=2006
    > ADG ArrDelay >0
```

```
     > GROUP BY Year,Month
```

```
SELECT Year,Month ,AVG(ArrDelay) AS AVG_arr, AVG(DepDelay) AS
     > AVG_dep
     > FROM airline_delay
     > WHERE delayYear=2006
     > AND delayYear=2007
     > GROUP BY Year,Month
     > ORDER BY Year,Month;
```

자바와 하이브를 연동하기위한 라이브러리필요

-

```
CREATE TABLE carrier_code(Code String ,Description String)
     > ROW FORMAT DELIMITED
     > FIELDS TERMINATED BY ','
     > LINES TERMINATED BY '₩n'
     > Stored as textfile;
```

```
load data local inpath '/root/airline/carriers.csv'
overwrite into table carrier_code
```

```
SELECT A.Year, A.UniqueCarrier, B.Code, B.Description
FROM airline_delay A
JOIN carrier_code B
ON (A.UniqueCarrier =B.Code)
Where A.delayYear =2006
LIMIT 20;
```

```
find . -name carriers.csv -exec perl -p -i -e 's/"//g' {} ₩;
```

“”제거해라

수행평가

1 데이터취합

2 Table 생성

3 Data Load

4 hive 데이터 분석

5 Java Application 데이터를 분석

```
create table city(rank int,city_code int,city_gu String)
ROW FORMAT DELIMITED
STORED AS TEXTFILE;
row format delimited fields   terminated by ',' stored as textfile;

LOAD DATA LOCAL INPATH '/root/cold/code.txt'
OVERWRITE INTO TABLE code;

create table code(city_code INT, city String) row format delimited fields   terminated by '          '
stored as textfile;

load data local inpath '/root/cold/code.txt' overwrite into table code;
```

hive --service hiveserver2 →얘가 떠있는상태에서 자바어플리케이션이 요청한다

문제들
한글문제, “”제거,숫자컴마,헤더정보 삭제,