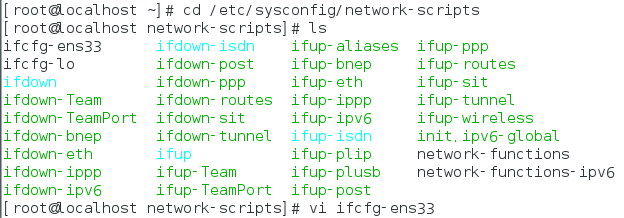
0. 편의를 위한 환경 설정 : (절전모드 해제) root > 설정 > 전원 > 빈 화면: 하지 않기 / 소프트웨어 업데이트 끄기(114p)

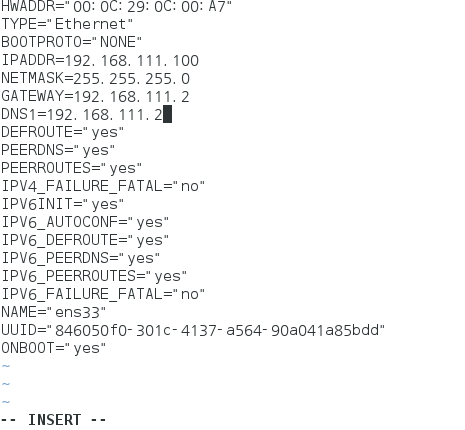
1. IP 주소 설정: 킬 때마다 변경되는 걸 고정시켜버리기 위함. (VMWare의 IP를 설정하고 CentOS의 IP를 고정 시킨 것)

경로는 아래.



<하둡 설치 파일 참고>

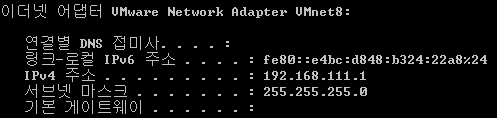
원래 파일을 오른쪽과 같이 수정 (리눅스 교재 67p)..



이후 책 68~69p 진행



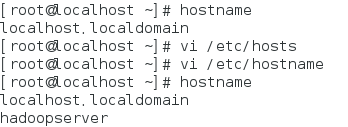
ipconfig 입력하면 VMnet8이 아래와 같다.





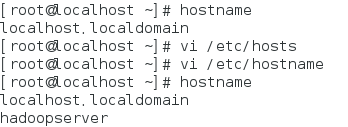
파이어폭스 열어서 인터넷 연결 잘되면 정상적인 것.

2. 하둡 서버 설정



vi /etc/hosts : 마지막 한 줄 추가

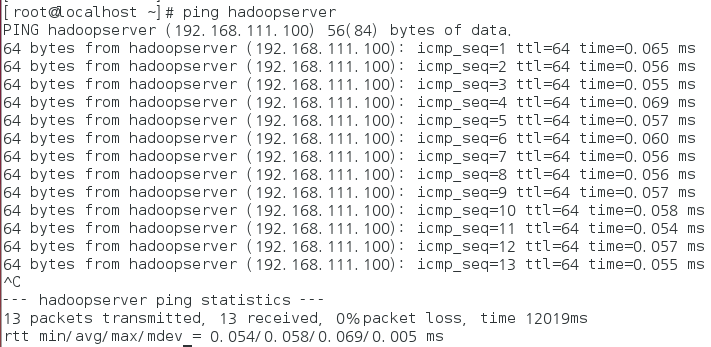




vi /etc/hostname : 변경

hadoopserver로 변경

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



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> /wget이라고 하면 스스로 url에 있는것받음.

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

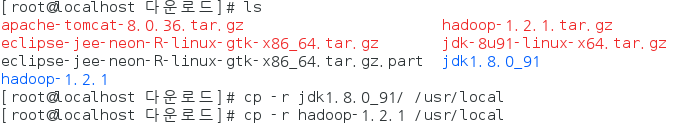
6. 이클립스 설치 : eclipse.org 가서 다운로드 > download packages 에서 Java EE 64bit 다운

7. 압축 풀기





8. 폴더 복사



9. 환경설정 파일 초기화



52라인에 아래와 같이 추가 (:set nu 로 줄 넘버 띄움)



**JAVA\_HOME=/usr/local/jdk-9.0.4**

**HADOOP\_HOME=/usr/local/hadoop-1.2.1**

**CLASSPATH=/usr/local/jdk-9.0.4/lib**

**export JAVA\_HOME HADOOP\_HOME CLASSPATH**

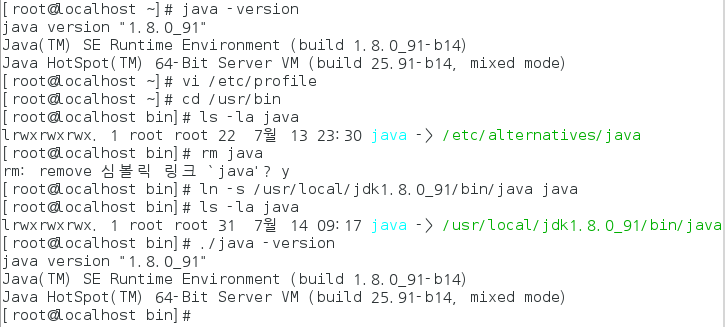
**PATH=$HADOOP\_HOME/bin:$JAVA\_HOME/bin:$PATH**

**wq하고**

**. /etc/profile 환경설정재가동**

**$PATH**

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



10. SSH

로컬> 하둡> conf가서,

ssh-keygen -t dsa -P '' -f ~/.ssh/id\_dsa :비밀번호 안치게하는 것! 보안푸는것.

(ssh server1 :내 자신으로 접속

yes

exit)

ls –a

[root@server1 ~]# cd .ssh

[root@server1 .ssh]# ls

id\_dsa id\_dsa.pub known\_hosts

((

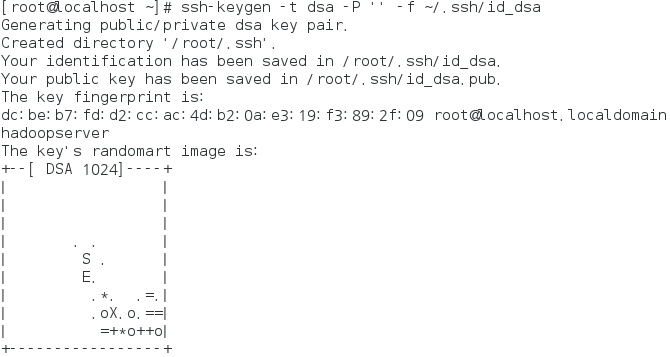
[root@server1 .ssh]# cat id\_dsa.pub >> authorized\_keys

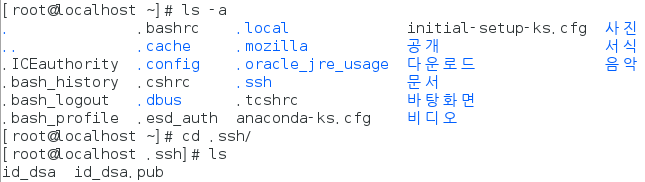
.ssh는 폴더임

잘안되면, 들어가서, rm –rf .ssh지워야함 하고 다시 만들어야함.

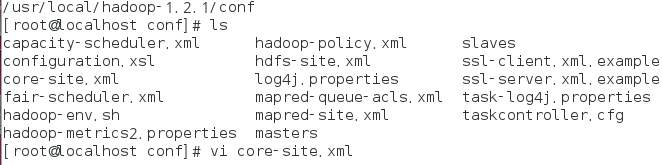
))

ssh-copy-id -i /root/.ssh/id\_dsa.pub root@server2 :server1에서 만든키값 2로전송.

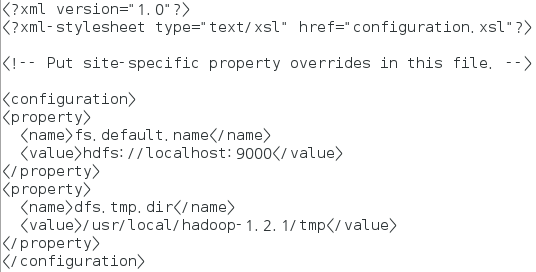








아래와 같이 추가



<property>

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

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

</property>

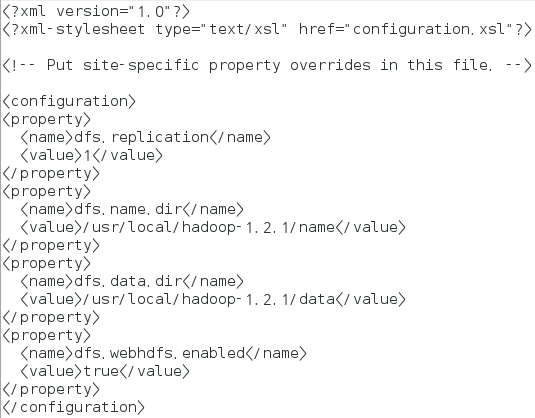
<property>

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

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

</property>

vi hdfs-site.xml 해서 아래와 같이 내용 추가



<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>

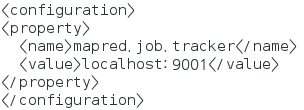
<name>dfs.webhdfs.enabled</name>

<value>true</value>

</property>

4yy 위로가서p

vi mapred-site.xml에 아래와 같이 추가



<property>

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

<value>localhost:9001</value>

</property>

ver1 conf]# systemctl stop firewalld

[root@server1 conf]# systemctl disable firewalld

하둡에 환경설정 파일

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

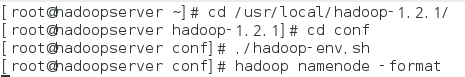


재부팅 함

hadoop치면 실행됨.

안되면 vi /etc/profile들어가서 제대로 적혀있는 지 확인!

이제 하둡 설치한 것~



아까 잘못 포맷하고 재시작해서 네임폴더가 만들어져서 제대로 안됨.

위에꺼 세번째줄 .뒤에 /없음!

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



파이어 폭스에 <http://localhost:50070> (192.168.111.100)입력하면 요렇게 뜨지요 얘는 모니터링용.

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

server1에서

start-all.sh

jps //현재동작되는 process check

stop-all.sh해놓고

아까 xml파일들 다 localhost 수정

localhost자리에 192.168.111.100으로 바꾸어주고, 50070

hdfs-site.xml

에는 4줄복사하고,

name 위에

name> dfs.http.address

value> localhost자리에 192.168.111.100으로 바꾸어주고, 50070

C:\Windows\System32\drivers\etc 에서 hosts파일을 word로 열어서

192.168.111.100 server1

맨아래 기입.. 다시하면됨1

다시 start-all.sh

후 크롬에서키면 됨!

cd /usr/local/hadoop-1.2.1

hadoop dfs -mkdir /test

하면 테스트라는 폴더생김 dfs는 하둡명령어를 쓰겠단말

데이터넣기:hadoop dfs -put README.txt /test

"리눅스 하둡에 있는 파일을 넣겠다!"

data폴더만들고,

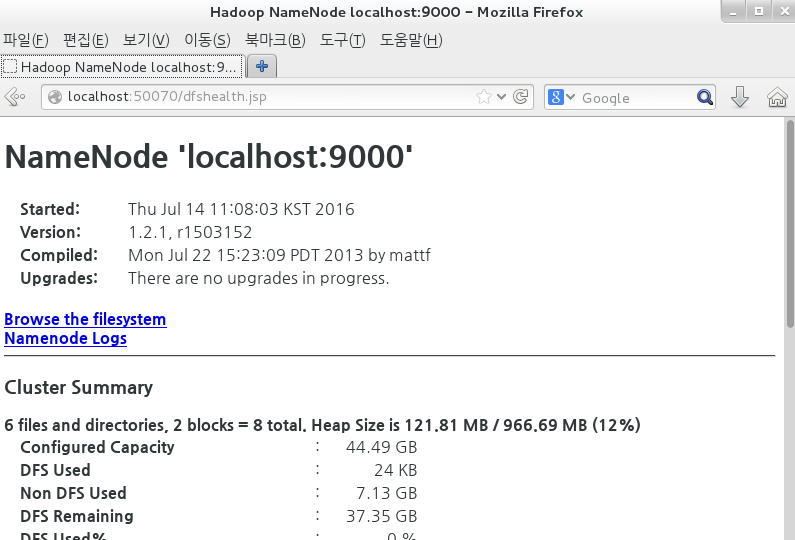
hadoop jar hadoop-examples-1.2.1.jar wordcount /test /result

test에 있는 readme.txt를 읽어서 /result라는 폴더에 결과를 만들어라.

result 폴더에 part-0000은 단어수 센거.

cd src/examples/org/apach/hadoop/examples

vi WordCount.java -->이게 아까 카운트 센것! 맵리듀스



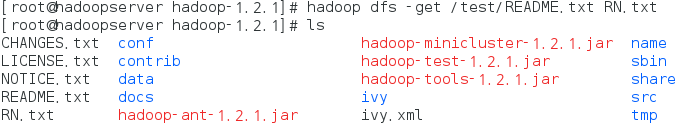
하둡시스템에 / (루트 디렉터리)의 내용을 보자는 내용.

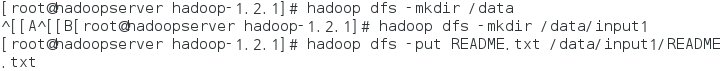


Test 라는 디렉토리를 만들어서 리드미를 거기에 올리고 가져오는 과정

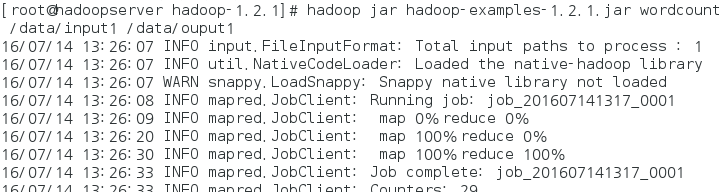




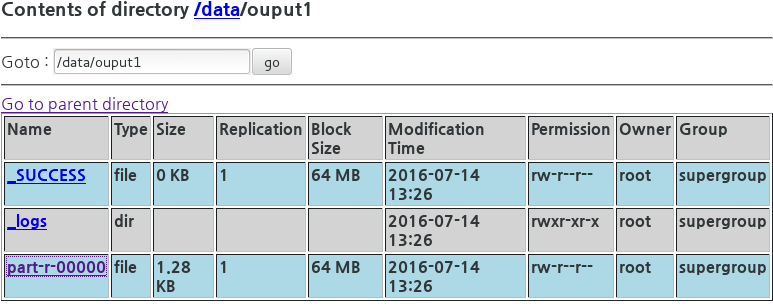


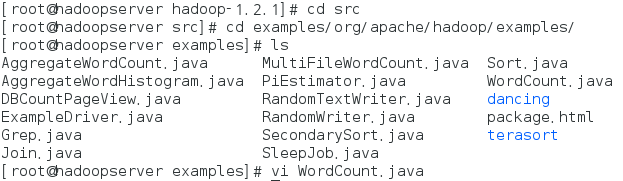


Input1 폴더의 파일들의 워드를 세어서 output1에 저장. hadoop~.jar는 자바 어플리케이션 역할



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





Part-r-0000은 지금 단어수세는거!

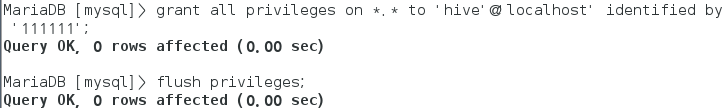
HIVE 설치

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

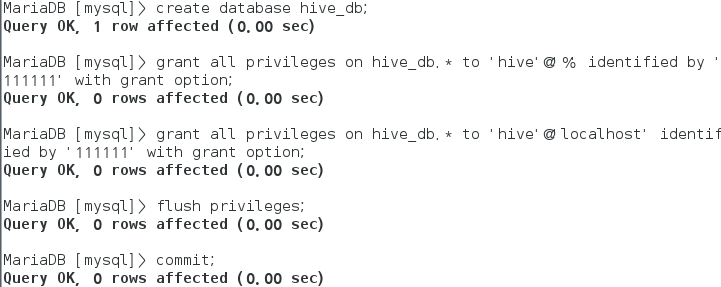
(<http://ftp.kaist.ac.kr/mariadb/mariadb-10.0.26/yum/centos7-amd64/rpms/>)



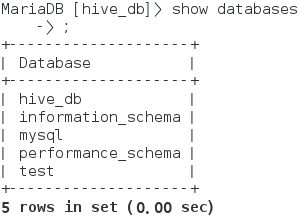
로컬호스트에서 접속하는 hive에게 권한을 다 주겠다.



hive\_db를 만들고 그 권한을 hive에게 다 줌







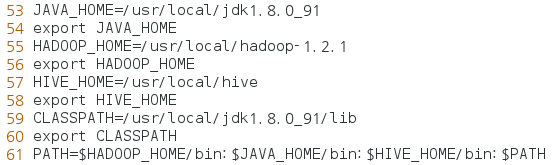
mysql 작업 끝

HIVE 세팅 시작합니다.

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



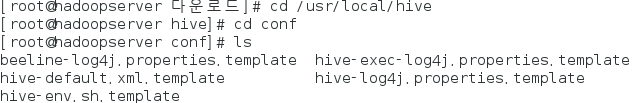
HIVE\_HOME 추가 57,58라인 추가하고 61라인에 패쓰에 하이브홈을 넣어줌











Hive-site.xml 만들고 메모장에 있는 내용을 넣어준다.



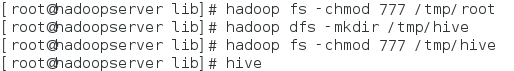
MariaDB JDBC API를 받아준다. 하이브 lib 안으로 옮겨줌



하이브가 하둡에서 활동하는 디렉토리를 만들어줌-> 하이브와 하둡 연결



Writable 할 수 있게 권한 수정



hadoop fs -mkdir /tmp

hadoop fs -mkdir /user/root/warehouse

hadoop fs -chmod 777 /tmp

hadoop fs -chmod 777 /user/root/warehouse

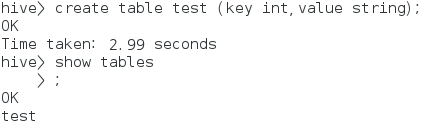
>hive 실행

-- error 후 tmp/hive 자동 생성

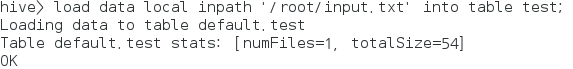
hadoop fs -chmod 777 /tmp/hive

재부팅합시다.

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

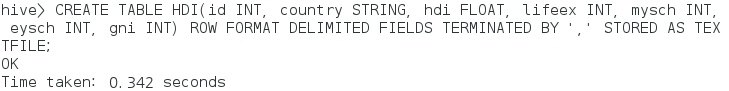


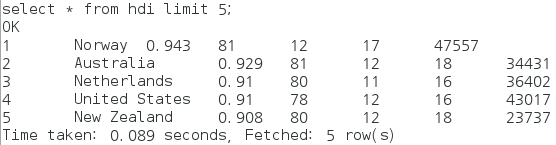
아래는 빅데이터 파일을 하이브를 통해서 하둡에 넣은 것이다.



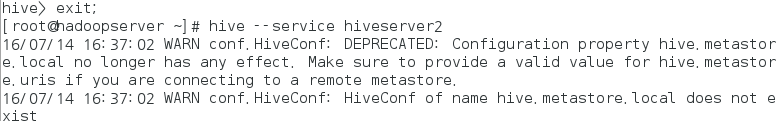
이거 말고 hdi-data.csv 파일 생성하고 테이블을 생성해서 넣으면 아래와 같이 테이블이 형성된다.



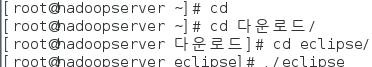




자바와 하이브가 연동될 수 있게 함.



이클립스 실행



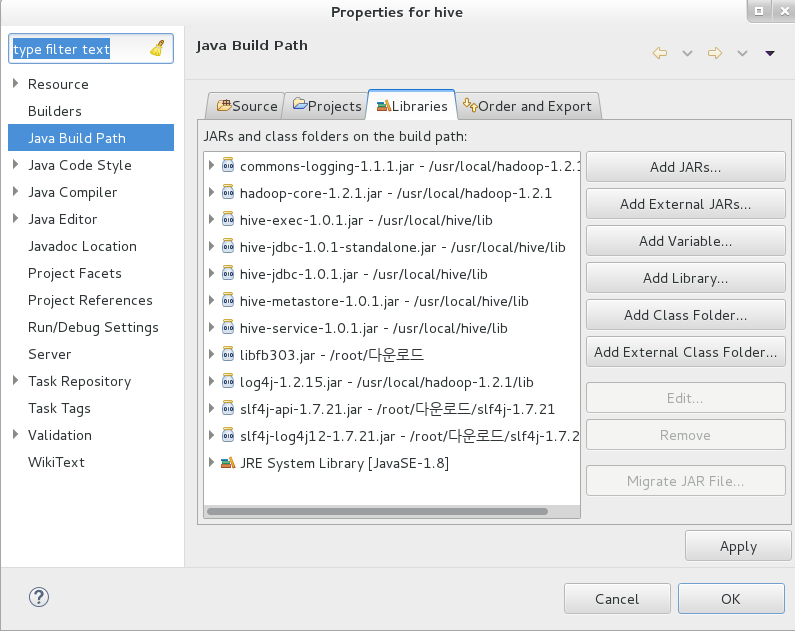
라이브러리 다운받기

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

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



이클립스 프로퍼티에 라이브러리 추가해주기



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....");

}

}