



하둡 환경 구축

zelratole@hanmail.net



하둡 환경구축 :

- 운영체제(리눅스)
 - master, slave1, slave2
- 설치 프로그램
 - jdk
 - hadoop
- 기타 툴
 - ssh : 서버와 클라이언트 시스템 간의 파일 전송처리
 - key, value 암호화 파일 전송
 - scp : 데이터 전송, 파일 전송



네트워크 구축 계획 :

- 분산 환경 구축
 - master, slave1, slave2
 - vagrant 툴 활용 :
<https://www.vagrantup.com/>
 - <https://releases.hashicorp.com/vagrant/1.9.6/>
 - [vagrant 1.9.6 x86_64.msi](#)
 - virtual box의 설치
 - <https://www.virtualbox.org/>
 - c:\bigdata
 - 설정 파일 : Vagrantfile 파일로 분산환경 설정
 - cmd 명령어



Vagrantfile 분산 환경 설정 :

- `nodes = {}` : 실제 분산 환경에서 처리할 ip 주소 매핑
 - 10.1.1.200, 10.1.1.201, 10.1.1.202
 - ex) `nodes = { 'master' ==> [1,200], 'slave1'==>[1, 201]....}`
 - 리눅스 system에 `/etc/hosts` 과 동일하게 처리..
 - 10.1.1.200 master
 - 10.1.1.201 slave
 - test: `ping slave`정상적으로 packet이 전송 확인



Vagrantfile 분산 환경 설정 :

- Virtualbox에 대한 선택
 - `config.vm.box = "centos/7"`
 - 가상 머신을 centos/7를 선택합니다.
- 사용자 계정 설정
 - `config.ssh.username="vagrant"` : root 외 사용자 계정을 설정
- 메모리와 cpu
 - `vbox.customize["modifyvm", : id, "--memory", 3074]`
 - `vbox.customize["modifyvm", : id, "--cpus",2]`



분산환경의 os 구동 처리.:

- 각 서버별 network ip setting
 - `sudo vi /etc/hosts`
 - 첫번째 줄 삭제(명령모드 dd)
 - i(insert모드)
 - `10.1.1.200 master master`
 - `10.1.1.201 slave1 slave1`
 - `10.1.1.202 slave2 slave2`
 - `esc :wq`
 - `sudo -i ==> passwd ==> 패스워드설정`
 - `systemctl restart network`
- 위 내용을 slave1, slave2에 동시 작업
- 각 서버에서 `ping slave1` 입력시, 정상적으로 데이터를 받은 것을 확인 할 때, 일단 분산 처리 완산



jdk 설치 :

- jdk
 - `sudo yum install java-1.8.0-openjdk-devel.x86_64`
 - 확인 : `java, javac`
 - `cd /usr/lib/jvm/java-1.8.0-[탭]`
- 환경 setting(vagrant)
 - `vi ~/.bashrc` 최하단
 - `export PATH`
 - `export JAVA_HOME=`

■ 설정 코드

```
ssh-keygen -t dsa -P " " -f ~/.ssh/id_dsa  
cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys  
chmod 644 ~/.ssh/authorized_keys  
ssh localhost
```




하둡설치..

- 하둡 설치(master만)
 - cd ~
 - wget : `sudo yum install wget`
 - wget
`http://ftp.daumkakao.com/apache/hadoop/common/hadoop-2.6.1/hadoop-2.6.1.tar.gz`
 - tar xvfz hadoop-2.6.1.tar.gz
 - ln -s hadoop-2.6.1 hadoop



하둡 환경 setting :

- 환경 세팅

- `sudo vi ~/.bashrc` 에 가장 하단

```
export HADOOP_PREFIX="$HOME/hadoop"  
export PATH=$PATH:$HADOOP_PREFIX/bin  
export PATH=$PATH:$HADOOP_PREFIX/sbin  
export HADOOP_MAPRED_HOME=${HADOOP_PREFIX}  
export HADOOP_COMMON_HOME=${HADOOP_PREFIX}  
export HADOOP_HDFS_HOME=${HADOOP_PREFIX}  
export YARN_HOME=${HADOOP_PREFIX}
```

- `source ~/.bashrc`
- 확인1 : `echo ${HADOOP_PREFIX}`
- 확인2 : `cd ${HADOOP_PREFIX}`
- `ls -l; pwd`



하둡 시작시 변경 파일 :

- `/home/vagrant/hadoop/etc/hadoop`
 - `hadoop-env.sh` : java홈, 하둡 lib, 컴파일 옵션.
 - `yarn-env.sh` : 하둡lib, 컴파일 옵션
 - `core-site.xml`
 - `hdfs://master:9000`
 - `hdfs-site.xml` : 하둡 파일 시스템 설정
 - namenode의 디렉토리 위치(이름)
 - namenode의 디렉토리 위치(데이터)
 - 권한설정
 - 복제옵션
 - `mapred-site.xml` : 프레임워크(yern)
 - map의 시스템 디렉토리 설정
 - map의 로컬 디렉토리 설정



- yarn-site.xml : 매투스 셔플 설정 옵션



hadoop-env.sh :

- vi ~/hadoop/etc/hadoop/hadoop-env.sh
 - 자바 홈 설정
 - 하둡의 라이브러리 위치 설정
 - 하둡 옵션으로 하둡 lib 위치 path

```
export JAVA_HOME=$JAVA_HOME
export
HADOOP_COMMON_LIB_NATIVE_DIR=${HADOOP_PREFIX}/lib
/native
export
HADOOP_OPTS="Djava.library.path=$HADOOP_PREFIX/lib"
```



yarn-env.sh

- vi ~/hadoop/etc/hadoop/yarn-env.sh
 - 하둡의 공통 lib 디렉토리
 - 하둡 옵션 lib path

```
export  
HADOOP_COMMON_LIB_NATIVE_DIR=${HADOOP_PREFIX}/lib/native  
export HADOOP_OPTS="-  
Djava.library.path=${HADOOP_PREFIX}/lib}"
```



- vi ~/hadoop/etc/hadoop/**core**-site.xml

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://master:9000</value>
    <final>true</final>
  </property>
</configuration>
```



- vi ~/hadoop/etc/hadoop/hdfs-site.xml
 - 파일시스템의 이름과 데이터, 권한여부, 복제 처리..

```
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:/home/vagrant/hadoop/dfs/name</value>
  <final>true</final>
</property>
<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:/home/vagrant/hadoop/dfs/data</value>
  <final>true</final>
</property>
<property>
  <name>dfs.permissions</name>
  <value>>false</name>
</property>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
```




mapred-site.xml :

- 설정내용
 - 맵리듀스 프레임웍의 이름 : yarn
 - 맵의 시스템 디렉토리 설정
 - 맵의 로컬 디렉토리 설정

```
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
<property>
  <name>mapred.system.dir</name>
  <value>file:/home/vagrant/mapred/system</value>
  <final>true</final>
</property>
<property>
  <name>mapred.local.dir</name>
  <value>file:/home/vagrant/mapred/local</value>
  <final>true</final>
</property>
```



yarn-site.xml :

- vi \$HADOOP_PREFIX/etc/hadoop/yarn-site.xml
 - 서버 관련 환경 setting, 핸들러 클래스 setting

```
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>

<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
```



- `hdfs namenode -format`
- `start-dfs.sh <--> stop-dfs.sh`
 - `http://10.1.1.200:50070`
- `start-yarn.sh <--> stop-dfs.sh`
 - `http://10.1.1.200:8088`
- 한 번 에
 - `start-all.sh <--> stop-all.sh`



dfs를 통한 명령처리 확인 :

- hdfs(하둡의 파일시스템)의 명령어를 통해서 data 노드의 폴드와 파일을 생성..
 - `hdfs dfs -mkdir /user`



보안키를 통한 접속 :

- ssh 보안
 - master와 slave1,. 간의 명령 및 파일 전송..
 - master에서 보안키를 생성, 각 node에 이 보안키를 전송하여, master가 각 node데이터 서버에 접속할 때, 인증과정을 거치지 않고도 ssh 명령을 수행 처리 가능..
- 처리 절차
 - master의 ~/.ssh/authorized_keys 파일의 내용에 보안 키 부분을 copy 한다.
 - slave1에 접속하여.. ~/.ssh/authorized_keys에 접속하여 위 master 부분을 copy 한다.
 - master에서 ssh로 slave1에 접속
 - \$ssh slave1
 - slave1에서 명령어를 수행하고 exit명령어를 통해 접속을 종료한다.



보안키 사용시 주의 점 :

- 보안 키는 1line으로 되어 있기 때문에 copy시에 정상적이지 않는 줄바꿈은 원래로 붙여준다..
- 연습
 - slave2에 jdk 설치, 보안키를 copy 해서 정상접속 여부 확인



보안키 설정 각 **node**에 파일전송 :

- master에서 각 노드별로 파일전송..
 - vagrant의 초기 위치 : `cd ~`
 - hadoop 파일을 전송
 - `scp -drp hadoop-2.6.1 slave1:~/`
 - `scp ~/.bashrc slave1:~/.bashrc`
- 확인예제
 - `slave2`에도 동일하게 적용 처리..



GUI 환경 setting :

- vagrant 시작(master)
 - `sudo yum -y groups install "GNOME Desktop"`
- master shutdown
 - `shutdown -h now`
- virtualbox로 실행
 - `vagrant//vagrant`
 - `startx`
- vagrant 설정 파일변경
 - `vb.gui=true`
- 재부팅 : `vagrant ssh master`

startx



- eclipse 설치..
 - 다운시, 반드시 상세 eclipse j2ee 통합개발 버전
 - 실행시, 권한 부분 문제일시
 - `sudo ./eclipse`

- local 파일 실행
 - conf 폴드 설정 없이 수행 eclipse에서 경로 지정으로 실행 가능
- hdfs 시스템 파일 실행
 - conf폴드 만들기
 - hadoop의 etc하위 설정 관련 xml 파일 복사
 - 작성 후, export @@@.jar
 - hadoop jar @@@.jar 패키지명.클래스명 argument값
 - hadoop jar wordcount.jar hadoopexp.WordCount input output01



- hdfs에서 출력
 - java code로 경로 설정
- eclipse java에서 처리할 부분
 - conf 폴드 생성
 - class 등록 처리..
 - conf 설정(xml)
 - 하둡에서 etc 하위 설정관련된 폴드에서 복사..



하둡 파일 시스템 빅데이터 처리 :

- hdfs에서 입력파일
 - 입력 폴드 만들기
 - touch data01.txt
 - vi data01.txt ==> himan! good job!!
 - hdfs dfs -mkdir /user/vagrant/input
 - hdfs dfs -put data01.txt /user/vagrant/input



Thank You !

www.themegallery.com