

자바쌤~!

https://cafe.naver.com/javassem

검색

63기-자바 & 빅데이터 >

[설치] 하둡



자바쌤 1:1 채팅
2020.06.22. 12:15 조회 0

댓글 0 URL 복사

JDK, Hadoop설치!!

0. Master, Slave 역할

nn01 = Master (NameNode, SecondaryNamenode, ResourceManager)

dn01 = Slave (DataNode, NodeManager)

dn02 = Slave (DataNode, NodeManager)

nn01, dn01, dn02 각각에서 JDK의 압축을 해제하고 설치하고 설정

moabTerm으로 반복 동작을 줄일 수 있다.

모든 사용자를 위한 java를 설치할 예정이니, root 계정으로 로그인한다.

설치되었는지 확인하기

```
[root@nn01 ~]$ which java
```

1. protobuf 설치

구글에서 공개한 오픈소스 직렬화 라이브러리

프로토콜 버퍼는 데이터를 연속된 비트로 만들고, 이렇게 만들어진 비트를 해석해 원래의 데이터를 만들 수도 있다. 현재 다양한 시스템이 이기종 혹은 내부 프로세스 간의 통신에 프로토콜 버퍼를 사용하고 있으며, 하둡2도 내부 데몬 간의 데이터 통신을 위해 프로토콜 버퍼를 적용했다.

(도서 '시작하세요 하둡프로그래밍'에서 언급하긴 했지만 설치 없어도 우리가 구동하는 것에는 문제가 되지 않았지만 혹시 몰라 설치해본다)

(1)설치전 필요 툴 (root 계정에서)

```
$ yum install -y autoconf automake libtool curl gcc-c++ unzip
```

(2) 다운로드 (책마다 버전이 중요하다고 강조하는데)

```
# cd /tmp
```

```
# wget https://github.com/protocolbuffers/protobuf/releases/download/v2.5.0/protobuf-2.5.0.tar.gz
```

```
# tar -zxvf protobuf-2.5.0.tar.gz
```

```
# mv protobuf-2.5.0 /opt/
```

이동되었는지 확인

(3) 설치 (protobuf 폴더로 이동)

```
# ./configure
```

```
# make ( 여기서 에러가 발생하면 ./configure에서 안되는 것임 )
```

```
# make install
```

(4)확인

```
protoc --version
```

만일 에러 발생시 Ifconfig (소문자 엘) 후 다시 protoc --version

2. JDK 8 설치 (nn01,dn01,dn02)

a. cd /tmp

b. yum install -y vim wget unzip

c. wget --no-check-certificate --no-cookies - --header "Cookie: oraclelicense=accept-securebackup-cookie"

<http://download.oracle.com/otn-pub/java/jdk/8u131-b11/d54c1d3a095b4ff2b6607d096fa80163/jdk-8u131-linux-x64.tar.gz>

(Local에 이미 다운로드 한 경우 Winscp를 이용하여 업로드)

d. ls jdk*

e. tar -xvzpf jdk-8u131-linux-x64.tar.gz

f. mkdir -p /opt/jdk/1.8.0_131

g. mv jdk1.8.0_131/* /opt/jdk/1.8.0_131/

카페정보 나의활동



매니저 자바쌤
since 2013.03.15.
카페소개

새싹2단계

437

초대하기

즐거찾는 멤버

64명

게시판 구독수

3회

우리카페엔 수

0회

카페 글쓰기

카페 채팅

즐거찾는 게시판

전체글보기

4,020

KOSMO

63기-자바 & 빅데이터

팁

- 심심문제
- 기타등등
- JAVA
- Oracle/Mysql
- JSP
- WebUI
- 파이썬
- Spring
- 리눅스
- 하둡+eco
- 프로젝트

공유해요

- 알고리즘도전
- 기술면접 준비하기
- 실존 면접
- 기회는 준비된자에게
- 프로젝트 팀

쉬어가기

청년아카데미

전문가과정

공지사항

카페회칙

카페탈퇴하기

- h. ln -s /opt/jdk/1.8.0_131 /opt/jdk/current
- i. install java with alternatives (예러)

alternatives명령어 : centos의 yum을 통해 java를 install하게 되면 버전관리 대상으로 들어간다.
그리고 centos는 버전관리를 위한 명령어를 제공하는데 그것이 바로 alternatives라는 명령어이다.

```
alternatives --install /usr/bin/java java /opt/jdk/1.8.0_131/bin/java 2
alternatives --config java
There is 1 program that provides 'java'.
Selection Command
-----
*+ 1 /opt/jdk1.8.0_131/bin/java
Enter to keep the current selection[+], or type selection number:1 (입력)
javac와 jar 명령어 경로도 alternatives 적용 권장
At this point JAVA 8 has been successfully installed on your system.
We also recommend to setup javac and jar commands path using alternatives
alternatives --install /usr/bin/jar jar /opt/jdk/1.8.0_131/bin/jar 2
alternatives --install /usr/bin/javac javac /opt/jdk/1.8.0_131/bin/javac 2
alternatives --set jar /opt/jdk/1.8.0_131/bin/jar
alternatives --set javac /opt/jdk/1.8.0_131/bin/javac
java -version
```

- 3. Hadoop 설치 (nn01 / dn01 / dn02)
 - a. cd /tmp
 - b. wget <http://apache.tt.co.kr/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz>
 - c. tar -xvzf hadoop-2.7.7.tar.gz
 - d. mkdir -p /opt/hadoop/2.7.7
 - e. mv hadoop-2.7.7/* /opt/hadoop/2.7.7/
 - f. ln -s /opt/hadoop/2.7.7 /opt/hadoop/current
- 4. Hadoop 사용자 추가
 - a. useradd hadoop
 - b. passwd hadoop (비밀번호도 hadoop 으로 - 비밀번호 입력시는 글자표시 안됨)
 - c. chown -R hadoop:hadoop /opt/hadoop/ (루트에서 만든 파일의 권한을 hadoop에게 권한)
 - d. su - hadoop

- 5. 자바 및 Hadoop 환경 변수 추가

(설정파일 수정할 때는 MultiExe 보다 하나씩 하는 것이 나을 수도 있다)

 - a. vi ~/.bash_profile


```
#### HADOOP 2.7.7 start #####
PATH=$PATH:$HOME/bin
export HADOOP_HOME=/opt/hadoop/current
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
#### HADOOP 2.7.7end#####
#### JAVA 1.8.0 start#####
export JAVA_HOME=/opt/jdk/current
export PATH=$PATH:$JAVA_HOME/bin
#### JAVA 1.8.0 end#####
```

b.source ~/.bash_profile

- 9. 자바 및 hadoop 버전 확인
 - a. java -version
 - b. hadoop version

- 10. 비밀번호없이 각노드를 접속할 수 있도록 공개키 공유(SSH)

(0) 아래부분은 가상머신에서 직접해야 한다. (*****)

 - vi 강제 종료 :q!
 - vi /etc/hosts (root 권한으로) - localhost 꼭 지운다 (모든 노드 다)

```
192.168.56.101 nn01
192.168.56.102 dn01
192.168.56.103 dn02
```

```

(1) 키만들기 (Hadoop 계정에서 )
( MultiExec 에서 하면 다른 키값을 생성되는 것을 볼 수 있다 )
[hadoop@nn01 ~]$ ssh-keygen
[hadoop@dn01 ~]$ ssh-keygen
[hadoop@dn02 ~]$ ssh-keygen
엔터만 3번친다
2.키복사하기
( 여기서는 MultiExec 풀고 작업 - 각 노드별로 작업)
( 자기 자신에게도 복사를 해야 됨 )
[hadoop@nn01 ~]$ ssh-copy-id hadoop@dn01
[hadoop@nn01 ~]$ ssh-copy-id hadoop@nn01
[hadoop@nn01 ~]$ ssh-copy-id hadoop@dn02

[hadoop@dn01 ~]$ ssh-copy-id hadoop@dn01
[hadoop@dn01 ~]$ ssh-copy-id hadoop@nn01
[hadoop@dn01 ~]$ ssh-copy-id hadoop@dn02
[hadoop@dn02 ~]$ ssh-copy-id hadoop@dn01
[hadoop@dn02 ~]$ ssh-copy-id hadoop@nn01
[hadoop@dn02 ~]$ ssh-copy-id hadoop@dn02
yes > 비밀번호 hadoop
-----
****패스워드없이 이동이 가능하다. ( 나올 때는 exit 또는 logout 으로 나온다 )
[hadoop@nn01 ~]$ ssh dn01
[hadoop@nn01 ~]$ ssh dn02
[hadoop@dn01 ~]$ ssh nn01
[hadoop@dn01 ~]$ ssh dn02
[hadoop@dn02 ~]$ ssh nn01
[hadoop@dn02 ~]$ ssh dn01

*****hadoop설정
nn01에서 하고 dn01, dn02 전송하던가
한 노드씩 같이 하던가
12. Hadoop 설정
( 설정은 MultiExec를 풀고 각 노드마다 하는 것이 낫다
또 복사하고 제발 확인하고 다시 열어서 또 확인하자
우선 nn01에서 작업하자
)
a. [hadoop@nn01 ~]$ vi /opt/hadoop/current/etc/hadoop/core-site.xml
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://nn01:9000</value>
</property>
</configuration>
-----
b. [hadoop@nn01 ~]$ vi /opt/hadoop/current/etc/hadoop/hdfs-site.xml
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.http-address</name>
<value>nn01:50070</value>
</property>
<property>
<name>dfs.namenode.secondary.http-address</name>
<value>nn01:50090</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/home/hadoop/hadoop_data/hdfs/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>

```

```
<name>dfs.datanode.data.dir</name>
<value>file:/home/hadoop/hadoop_data/hdfs/datanode</value>
</property>
<property>
<name>dfs.namenode.checkpoint.dir</name>
<value>file:/home/hadoop/hadoop_data/hdfs/namesecondary</value>
</property>
<property>
<name>dfs.webhdfs.enabled</name>
<value>true</value>
</property>
</configuration>
```

c. [hadoop@nn01 ~]\$ vi /opt/hadoop/current/etc/hadoop/yarn-site.xml

```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>nn01:8030</value>
</property>
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>nn01:8031</value>
</property>
<property>
<name>yarn.resourcemanager.address</name>
<value>nn01:8032</value>
</property>
<property>
<name>yarn.resourcemanager.hostname</name>
<value>nn01</value>
</property>
</configuration>
```

d. [hadoop@nn01 ~]\$ cp /opt/hadoop/current/etc/hadoop/mapred-site.xml.template
/opt/hadoop/current/etc/hadoop/mapred-site.xml

e. [hadoop@nn01 ~]\$ vi /opt/hadoop/current/etc/hadoop/mapred-site.xml

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
<property>
<name>mapreduce.jobtracker.hosts.exclude.filename</name>
<value>${HADOOP_HOME}/etc/hadoop/exclude</value>
</property>
<property>
<name>mapreduce.jobtracker.hosts.filename</name>
<value>${HADOOP_HOME}/etc/hadoop/include</value>
</property>
</configuration>
```

f. [hadoop@nn01 ~]\$ vi /opt/hadoop/current/etc/hadoop/masters
nn01

g. [hadoop@nn01 ~]\$ vi /opt/hadoop/current/etc/hadoop/slaves
dn01
dn02

```
-----
h. [hadoop@nn01 ~]$ vi /opt/hadoop/current/etc/hadoop/hadoop-env.sh
# The java implementation to use.
export JAVA_HOME=/opt/jdk/current
ii. [hadoop@nn01 ~]$ vi /opt/hadoop/current/etc/hadoop/yarn-env.sh
# some Java parameters
export JAVA_HOME=/opt/jdk/current
```

ModaXterm으로 작업하지 않았으면 아래부분 작업으로 해도 된다

```
=====
hadoop설정을 nn01에서하고 dn01과 dn02에 복사 (winscp툴을 이용하거나 scp명령을 이용한다.
혹은
[nn01(root 계정)]
a. [root@nn01 ~]# scp -r /opt/hadoop/* dn01:/opt/hadoop
a. [root@nn01 ~]# scp -r /opt/hadoop/* dn02:/opt/hadoop
[dn01(root 계정)] 심볼릭링크와 소유자를 다시 설정한다.
a. [root@dn01 ~]# rm -rf /opt/hadoop/current
b. [root@dn01 ~]# ln -s /opt/hadoop/2.7.7 /opt/hadoop/current
c. [root@dn01 ~]# chown -R hadoop:hadoop /opt/hadoop/
[dn02(root 계정)] 심볼릭링크와 소유자를 다시 설정한다.
a. [root@dn02 ~]# rm -rf /opt/hadoop/current
b. [root@dn02 ~]# ln -s /opt/hadoop/2.7.7 /opt/hadoop/current
c. [root@dn02 ~]# chown -R hadoop:hadoop /opt/hadoop/
=====
[ 확인 ] ll /opt/hadoop
```

각노드에서 해야 하니까 MobaXterm 멀티창 끄기

```
14. Hadoop namenode 디렉토리 생성 (nn01 : Namenode)
a. mkdir -p ~/hadoop_data/hdfs/namenode
b. mkdir -p ~/hadoop_data/hdfs/namesecondary
15. Hadoop datanode 디렉토리 생성 (dn01 : Datanode)
mkdir -p ~/hadoop_data/hdfs/datanode
Hadoop datanode 디렉토리 생성 (dn02 : Datanode)
mkdir -p ~/hadoop_data/hdfs/datanode
```

```
16. Namenode 포맷 (nn01)
hadoop namenode -format
17. Daemon 시작 (nn01)
a. start-all.sh
```

```
18. 서비스 정상 확인
[hadoop@nn01 ~]$ jps
16624 ResourceManager
16883 Jps
16232 NameNode
16462 SecondaryNameNode
```

```
[hadoop@dn01 ~]$ jps
10518 DataNode
10650 NodeManager
10780 Jps
```

```
[hadoop@dn02 ~]$ jps
7041 DataNode
7141 NodeManager
7271 Jps
```

GUI (윈도우 브라우저로 접속가능)

<http://192.168.56.101:50070/>

<http://192.168.56.101:8088/>

오류찾기

```
[hadoop@nn01 current]$ ls -al /opt/hadoop/current/logs/
```

21. hdfs 명령어 테스트

```
[hadoop@nn01 ~]$
```

```
a. hdfs dfs -ls -R /
```

```
b. hdfs dfs -mkdir /input
```

```
c. hdfs dfs -put /opt/hadoop/current/NOTICE.txt /input
```

```
d. hdfs dfs -ls /input
```

```
e.
```

기존에 출력 폴더 있으면 에러 발생 : hdfs dfs -rm -r /output 확인

```
hadoop jar /opt/hadoop/current/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.7.jar wordcount /input /output
```

```
hdfs dfs -cat /output/part-r-00000/
```

[su 명령어로 계정변경 가능하도록] - 각 노드에서 모두 수정

```
root->vagrant->hadoop
```

```
hadoop에서 su - 가 인증실패됨
```

이를 해결하기위해

/etc/pam.d/su 파일의 vagrant포함되어 있는 부분 3줄삭제 또는 hadoop으로 변경

가상머신에서 root 계정에서 직접수정

```
[root@nn01 ~]# vi /etc/pam.d/su
```

10,11,12 라인에 vagrant 라고 보이는 3줄 주석처리

```
[root@nn01 ~]# su - hadoop //가능해짐
```

```
[hadoop@nn01 ~]
```

* sudo 명령어 안될 때

root 계정에서 visudo -f /etc/sudoers

ecs :100 라인 root ALL=(ALL) ALL 을 yy로 복사하고 바로 p 눌러서 붙이고

root를 hadoop으로 변경

```
root ALL=(ALL) ALL
```

```
hadoop ALL=(ALL) ALL
```

여기에 패스워드까지 묻지 않고 연결하고 싶으면

```
hadoop ALL=(ALL) NOPASSWD: ALL ( 계정 등록 )
```

```
%hadoop ALL=(ALL) NOPASSWD: ALL ( 그룹 등록 )
```

*** 나갈 때 제발 > stop-all.sh 꼭

댓글

댓글알림

김용선

댓글을 남겨보세요



등록

자바쌤~! <https://cafe.naver.com/javassem>