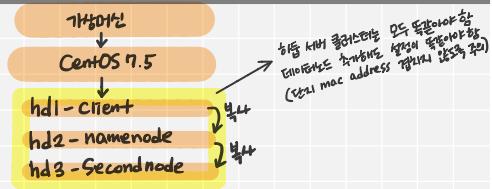


# 하둡 설치 (클러스터 모드)

- ① 가상머신 만들고 리눅스 CentOS 7.5 설치
- ② 1번 가상머신에 필요한 파일 다운받고 VDI 이미지로 불러온다.
- ③ 배포한 VDI 이미지를 가상머신 설치한 뒤 2, 3번도 넣는다.



## 가상머신 만들기 (hd1 - Client)

- OS: Red hat 6.4 버전
- OS: Linux
- ① 네트워크 설정
- ② 1번 어댑터 mac address 설정  
[기본값은 'eth0'이나 'eth1' 등]
- ③ 2번 어댑터 [ ] 네트워크 어댑터 사용 설정
- ④ 2번 어댑터 mac address 설정
- ⑤ 가상머신 시작
- ⑥ 로그인 ID: root PW: hadoop
- ⑦ \$ ip a ⇒ 아이피를 확인한다
- ⑧ \$ ifconfig enp0s8 192.168.56.21 ⇒ 아이피를 고정한다
- ⑨ \$ ip a ⇒ ottomn 192.168.56.21인지 확인한다

서버는 가상머신을 설치하는 IP 주소로 접속  
각각의 서버마다 다른 IP 주소로 접속  
각각의 서버마다 다른 IP 주소로 접속

## SSH 접속 (hd1 - Client)

- ① 터미널에서 txt 파일을 open (ppt에서는 '-'가 2자이므로 비워둠)
  - ② PuTTY 실행 Address: 192.168.56.21  
SSH 키
  - ③ Putty 경고창에 Putty Security Alert이 표시됨  
처음 연결하는 경우 Putty Security Alert이 표시됨.  
fingerPrint에 대한 정보가 표시됨.  
이에 무조건 'YES'로 선택함. 아래 경고 대화창 참조
  - ④ Putty SSH \$ hostname ⇒ 환경설정 확인 (Client.hadoop.kr)
  - ⑤ \$ ping -c 2 www.google.com ⇒ 네트워크 되는지 확인
- \* 아래부분은 환경설정 반드시 확인

수정을 방지하기 위한 설정 10개가 있음  
root : 자신의 기본설정  
etc/hosts  
etc/hostname  
  
hadoop : 6개의 설정파일 (2개는 slave)  
1번 1. 리눅스 환경  
2번 2. 환경설정 (환경)  
3번 3. 주기설정 (마스터설정)  
마스터로 부터 받은 환경설정  
hadoop설정은 환경설정이므로  
Slaves

## HOST 설정

- ① \$ cat /etc/hosts ⇒ 기본설정 확인
- ② \$ echo ① 192.168.56.21 Client.hadoop.kr Client' >> /etc/hosts  
echo ② 192.168.56.31 namenode.hadoop.kr namenode' >> /etc/hosts  
echo ③ 192.168.56.41 Secondnode.hadoop.kr Secondnode' >> /etc/hosts  
echo ④ 192.168.51.31 datanode1.hadoop.kr datanode1' >> /etc/hosts  
echo ⑤ 192.168.51.41 datanode2.hadoop.kr datanode2' >> /etc/hosts

클라이언트와 노드간의 것을 매핑

- ③ 다시 \$ cat /etc/hosts 터미널로 출력해보았다

## Hadoop 환경설정

- ① \$ su - hadoop
- ② \$ whoami ⇒ 터미널은 root 권한으로 실행되었던 것임  
↳ 'hadoop'으로는 실행못함
- ③ \$ pwd ⇒ /home/hadoop

## 리눅스 환경설정

- ① \$ cp .bashrc org\_.bashrc
- ② \$ cp bashrc2.txt .bashrc
- ③ \$ source .bashrc
- ④ \$ cat .bashrc
- ⑤ \$ env ⇒ 환경설정 파일 프로그램

## 리눅스 명령어

- \* Cat file1 : file1의 내용을 출력
- \* Cat file1 file2 file3 : file1, 2, 3의 내용을 연속 출력
- \* Su - 터미널 : 터미널을 강제로 로그인 한뒤 슈퍼유저 권한을 사용. Su - 만 명령줄을 사용하고 root 터미널
- \* Cp : 복사 (copy)
- \* Mv : 이동 (move)
- \* Pwd : 현재 작업장을 출력해 위치를 확인
- \* Source : 스크립트 파일 수동으로 실행시킬때 파일내(기계적 or 사용자인 환경변수), Source 명령어를 사용하여 사용환경을 비우거나 설정
- \* Rm : 파일, 디렉토리 삭제 (remove)
- \* Wget : 웹에서 파일다운로드

## JDK 버전 확인

- ① \$ javac -version → (java가 아니고 javac임) java - 끝에 java는 java - 끝에 java는  
javac - 가 끝에 java는  
② \$ jps → jvm 위에서 돌아가는 프로세스를 확인하는 툴

## SSH 설정

↳ 클라우스터의 X개의 컴퓨터에 접속하기 위해서 기상마신을 복제한 후 PW없이 SSH 접속이 가능하도록 해줌

- ① ssh localhost → 처음 사용할 때 비밀번호 입력 후 Enter 치면  
② ls -al .ssh → known\_hosts라는 파일을 보면 finger print 내용을 담고 있는 파일. 혹시라도 내용에 변이가 있다면 해당 파일을 지우고 다시 생성

③ rm .ssh/known\_hosts → known\_hosts 파일 초기화

## 하둡 설정

- ① hadoop-env.sh
- ② core-site.xml
- ③ hdfs-site.xml
- ④ mapred-site.xml
- ⑤ yarn-site.xml
- ⑥ log4j.properties
- ⑦ slaves

} 각각 모든 서버에 공통  
각각 다른 IP로 ppt

## 기상마신 복제하기

- ① \$ exit → 로그아웃  
② \$ whoami → root 대체 확인  
③ \$ shutdown -h now → 기상마신 종료 (Putty에서 DDoS)

당시 각각 인증서를 Virtual Box의 가상머신에서 [복사] 누르기 [기상마신에 붙여넣기]

## 여기까지 하둡 Client 설정

↳ 이제는 원격으로 Client를 실행해보자 사용

### [Virtual Box에서 진행]

- ① 1번 클러스터 (hd1-client)를 복제한다  
② 2번 클러스터 (hd2-namenode)와 3번 클러스터 (hd3-secondnode) 복제  
③ 모든 네트워크 카드의 MAC 주소 추가하기 네트워크 [V] / 연결된 네트워크 [v]  
④ 기상마신 3대 모두 시작  
⑤ hd1 기상마신은 뜨거운 IP 주소로 IP 고정설정 a. \$ ip a  
b. \$ ifconfig eth0:0 192.168.56.21/31/41  
c. \$ ip a

### [ip 고정설정 Putty에서 진행]

- ↳ 192.168.56.21  
① Putty.exe를 SSH 터미널로 고정한 IP 입력 후 접속 \* Putty는 SSH 접속할 hd1-client만 한다. 클라우스터 3대는 SSH 접속하지 않는다  
② Client는 SSH로 접속한 \$ hostname  
③ \$ echo 'ifconfig eth0:0 192.168.56.21' >> /etc/rc.d/rc.local  
↳ 브팅할 때마다 자동으로 스트립트 실행  
    ⇒ ifconfig ~'를 rc.local에다가 추가해도 좋지만 끊어짐  
    Ip를 ip3로 바꿔서 한다  
④ 하지만 각각의 설정파일을 재등록으로 반복문으로 실행해주는 편이 좋다  
    \$ chmod +x /etc/rc.d/rc.local

⑤ [hd2-namenode]는 서로 SSH에 접속하는게 아니라 client에서 접속한다!

### ⑥ \$ ssh namenode

⑦ \$ echo 'namenode.hadoop.kr' > /etc/hostname

⑧ \$ hostname namenode.hadoop.kr ⇒ 이름 변경되어 네임노드로 설정된다. 키워드로는 이름변경 X

### ⑨ \$ hostname

⑩ \$ ifconfig eth0:0 192.168.56.31 >>/etc/rc.d/rc.local

⑪ \$ chmod +x /etc/rc.d/rc.local

⑫ \$ W ⇒ 로그인한 사용자의 정보 티어

⑬ \$ exit

⑭ \$ ssh namenode

⑮ \$ exit

3번 클러스터는  
이동안 네트워크  
접속 설정!

비번 없이  
자동으로 접속  
(자동화 설정)

비밀번호없이 다른머신에  
접속하기 시나리오

## ※ 비번없이 서버에 접속해야 하므로 보안기술이 필요하다

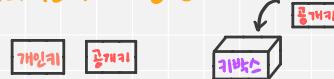
[ ssh계정으로 실행, root 계정으로 ]

↳ 하지만 늘 헛갈리므로 잘 알지않는 예전어 \$ (IP A) (상당 빵이 놓쳤지 아파아파)

WHY??

클라우드의 개수가 많아지고, 알아보기 힘든 비번의 개수가 늘어나고 관리를 어렵고, 클라우드를 관리할 때 모든 클라우드에 접속권한이 있으면 관리가 편해

### 보안기술의 원리 (가장 중요\*)



개인 key와 공개 key를 만들고 접속대상 머신의 키박스에 공개키를 등록

→ 개인 key - 공개 key : id\_dsa - id\_dsa.pub



→ 키박스에 다수의 공개키를 등록할 수 있음 : Authorized\_keys

↳ 등록해놓으면 비번없이 접속할 수 있다

∴ 보안기는 단방향 : 내가 서버에 접속하는 것, 서버가 나에게 접속하는게 X

ex)



저 물에 우리모두 들어가려면 우리의 공개키를 물에 달려있는 키박스에 넣어야해!  
개인키는 알려주면 보안이 안되니 개인키는 보안만 알고있는 걸로 해야지!

## R, W, X 권한

read (r) → 4  
write (w) → 2  
execute (x) → 1

x y z

보안 그룹 기타

ex) 4 0 0 ⇒ 보안에게는 읽는권한만 주고  
그룹과 기타에게는 어떠한  
권한도 없음

### [ hd1-client ]

① \$ ssh-keygen -t dsa -P "" -f ~/.ssh/id\_dsa ⇒ 개인키와 공개키 생성

② \$ cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys ⇒ 공개키를 키박스에 추가

③ \$ chmod 400 ~/.ssh/authorized\_keys ⇒ 접근권한 변경 (타인에게 노출되면 안되므로 읽기만 가능하도록 접근권한변경)

(↳ 암호화를 하면 개인이라도 쓰기권한을 박탈하고 읽기만 가능하게 해야한다)

④ ls -al .ssh

⑤ ssh localhost

⑥ w

⑦ exit

⑧ \$ scp ~/.ssh/id\_dsa.pub hadoop@namenode://home/hadoop/.ssh/client.pub

\$ Scp ~/.ssh/id\_dsa.pub hadoop@secondnode://home/hadoop/.ssh/client.pub

⇒ (본 머신(client)의 공개키를 2번, 다른 머신에 전송

## 간접 명령어

SSH-keygen -t dsa  
 : dsa 암호화방식으로 키를 생성한다 (ssh keygen은 기본적으로 RSA 방식)  
 -f : 저장할 파일명 지정, 경로 지정  
 |S - al : 파일명까지 복사  
 -P : 다른 포트 설정  
 SCP : Secure copy. 원격지에 있는 파일이나 폴더 영속으로 ssh를 통한 파일전송 방식

## [hd2-namenode]

- ① \$ ssh namenode
- ② \$ ssh-keygen -t dsa -P "" -f ~/.ssh/id\_dsa ⇒ 개인키와 공개키를 생성
- ③ \$ mv ~/.ssh/id\_dsa.Pub ~/.ssh/namenode.Pub ⇒ 공개키 이름을 서버하기 쉽게 변경
- ④ \$ SCP ~/.ssh/namenode.Pub hadoop@Secondnode: //home/hadoop/.ssh/namenode.Pub  
 ⇒ 공개키를 같은 클러스터(Secondnode)에 복사
- ⑤ \$ exit

↑ 4번과 5번은 2번

파일명  
Secondnode : //home/hadoop/.ssh/namenode.Pub  
Secondnode

경로명/파일명

## [hd3-Secondnode]

- ① \$ ssh Secondnode
- ② \$ ssh-keygen -t dsa -P "" -f ~/.ssh/id\_dsa ⇒ 개인키와 공개키를 생성
- ③ \$ mv ~/.ssh/id\_dsa.Pub ~/.ssh/Secondnode.Pub ⇒ 공개키 이름을 서버하기 쉽게 변경
- ④ \$ SCP ~/.ssh/Secondnode.Pub hadoop@namenode: //home/hadoop/.ssh/Secondnode.Pub  
 ⇒ 공개키를 같은 클러스터에 복사
- ⑤ \$ exit

## [hd2-namenode]

- ① \$ ssh namenode
- ② \$ ll .ssh
- ③ \$ Cat ~/.ssh/client.Pub >> ~/.ssh/authorized\_keys  
 Cat ~/.ssh/namenode.Pub >> ~/.ssh/authorized\_keys  
 Cat ~/.ssh/Secondnode.Pub >> ~/.ssh/authorized\_keys
- ④ \$ chmod 400 ~/.ssh/authorized\_keys
- ⑤ \$ exit

## [hd3-Secondnode]

- ① \$ ssh Secondnode
- ② \$ ll .ssh
- ③ \$ Cat ~/.ssh/client.Pub >> ~/.ssh/authorized\_keys  
 Cat ~/.ssh/namenode.Pub >> ~/.ssh/authorized\_keys  
 Cat ~/.ssh/Secondnode.Pub >> ~/.ssh/authorized\_keys
- ④ \$ chmod 400 ~/.ssh/authorized\_keys
- ⑤ \$ exit

## 확인작업

### [hd1-client]

2번과 3번 마シン으로 SSH 접속 테스트

- ① \$ ssh namenode
- ② \$ w
- ③ \$ exit
- ④ \$ ssh Secondnode
- ⑤ \$ w
- ⑥ \$ exit

### [hd2-namenode]

- ① \$ SSH namenode  
 ⇒ 먼저 2번 마シン으로 SSH 접속
- ② 4번과 3번 마シン으로 SSH 접속 테스트  
 \$ ssh localhost  
 \$ w  
 \$ exit  
 \$ ssh namenode  
 \$ w  
 \$ exit  
 \$ ssh Secondnode  
 \$ w  
 \$ exit
- ③ \$ exit ⇒ 2번마신에서 3번마신으로 테스트 Client2
- ④ \$ w

### [hd3-Secondnode]

- ① \$ SSH namenode  
 ⇒ 먼저 3번 마シン으로 SSH 접속
- ② 4번과 2번 마シン으로 SSH 접속 테스트  
 \$ ssh localhost  
 \$ w  
 \$ exit  
 \$ ssh namenode  
 \$ w  
 \$ exit  
 \$ ssh Secondnode  
 \$ w  
 \$ exit
- ③ \$ exit ⇒ 3번마신에서 2번마신으로 테스트 Client2
- ④ \$ w