0908

start-dfs.sh // 4개 뜨는지 확인

파이어폭스에서 <http://localhost:50070> 접속해보기

vm에서 설정 – 네트워크 – 포드포워딩 – 추가 – 호스트 포트,게스트 포트 50070으로, 나머지는 똑같이

firewall-cmd --zone=public --add-port=50070/tcp –permanent //방화벽개방

firewall-cmd --reload

윈도우에서 <http://192.168.10.1:50070> 접속

start-yarn.sh

jps

로컬에서 확인 : http://localhost:8088

stop-dfs.sh

jps

stop-yarn.sh

jps

start-dfs.sh

jps

hdfs dfs -mkdir /user // user 디렉토리 생성

hdfs dfs -ls / // user 디렉토리 생성 확인

hdfs dfs -mkdir /user/root

hdfs dfs -mkdir /user/centos

hdfs dfs -ls /user

hdfs dfs -mkdir /user/root/conf

hdfs dfs -ls -R / // recursive 옵션

ll

cd hadoop2

ll // README.txt 파일 사용할 것임 ~

현재 로컬에 있는 readme.txt 파일을 hdfs로 업로드

hdfs dfs -copyFromLocal $HADOOP\_HOME/README.txt /user

hdfs dfs -ls /user

hadoop jar $HADOOP\_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.10.0.jar wordcount /user/README.txt wordcount\_output

hdfs dfs -ls

hdfs dfs -ls wordcount\_output

hdfs dfs -cat wordcount\_output/part-r-00000 //cat으로 안의 내용 확인 가능

[완전 분산 모드]

master를 복제한다

hadooplab폴더에 slave1,2 완전한 복제

설정-시스템-1024GB로

마스터 시작- 오른쪽 위 시간 옆에 아이콘 – 설정 모양 – 네트워크 이더넷 2개 확인 (이름도 확인 enp0s3, enp0s8) 켬으로 설정 – 다시 터미널

cd /etc/sysconfig/network-scripts/

ll

cat ifcfg-enp0s3

gedit ifcfg-enp0s8 // 현재 파일이 없어서 만드는 것

DEVICE=enp0s8

ONBOOT=yes

BOOTPROTO=static

IPADDR=192.168.10.10

NETMASK=255.255.255.0 //추가하고 닫기

systemctl restart network

ifconfig

slave들도 동일하게 설정(ip만 다르게)

-ping 명령어로 잘 연결됐는지 확인

master에서

ping 192.168.10.11

중지는 ctrl+c

ping 192.168.10.12

각각 slave에서

ping 192.168.10.10

ping 192.168.10.12 등 자기가 아닌 ip확인

// 서로 잘 들리니~~~~~~~~? 통신이 되고 있니~~~~~~~~~~?

master에서

ssh 192.168.10.11 //yes 연결을한다

exit // 서버이름들이 기본값으로 loclahost라고 돼있어서 구분이 안됨, 바꿔 주도록 한다

각 guest os의 host명을 변경 : 내가 어디랑 연결, 작업하고 있는지 쉽게 확인가능

gedit /etc/host

127.0.0.1 localhost

192.168.10.10 master

192.168.10.11 slave1

192.168.10.12 slave2 // 원래 내용 다 지우고 추가, 저장 닫기

master에서

gedit /etc/hostname

master // 원래 내용 다 지우고 추가, 저장 닫기

/bin/hostname -F /etc/hostname

reboot

나머지 slave(1,2)도 동일하게 설정

---------------------------------------------------------------------------------------------------------------

인증키 전달

master에서

ssh slave1

exit

ssh slave2

exit

slave1에서

ssh master

exit

ssh slave2

exit

나머지 slave에서도 동일하게 테스트

만약 인증이 풀렸을 경우

r : recursive : 하위 디렉토리 및 파일을모두 전송

p : persistence : 원본파일 수정/사용시간 및 권한 유지함

scp -rp ~/.ssh/authorized\_keys >> root@slave1:~/.ssh/authorized\_keys

scp -rp ~/.ssh/authorized\_keys >> root@slave2:~/.ssh/authorized\_keys

scp -rp ~/.ssh/authorized\_keys >> root@slave3:~/.ssh/authorized\_keys