

Copy of 2020-02-12(수)

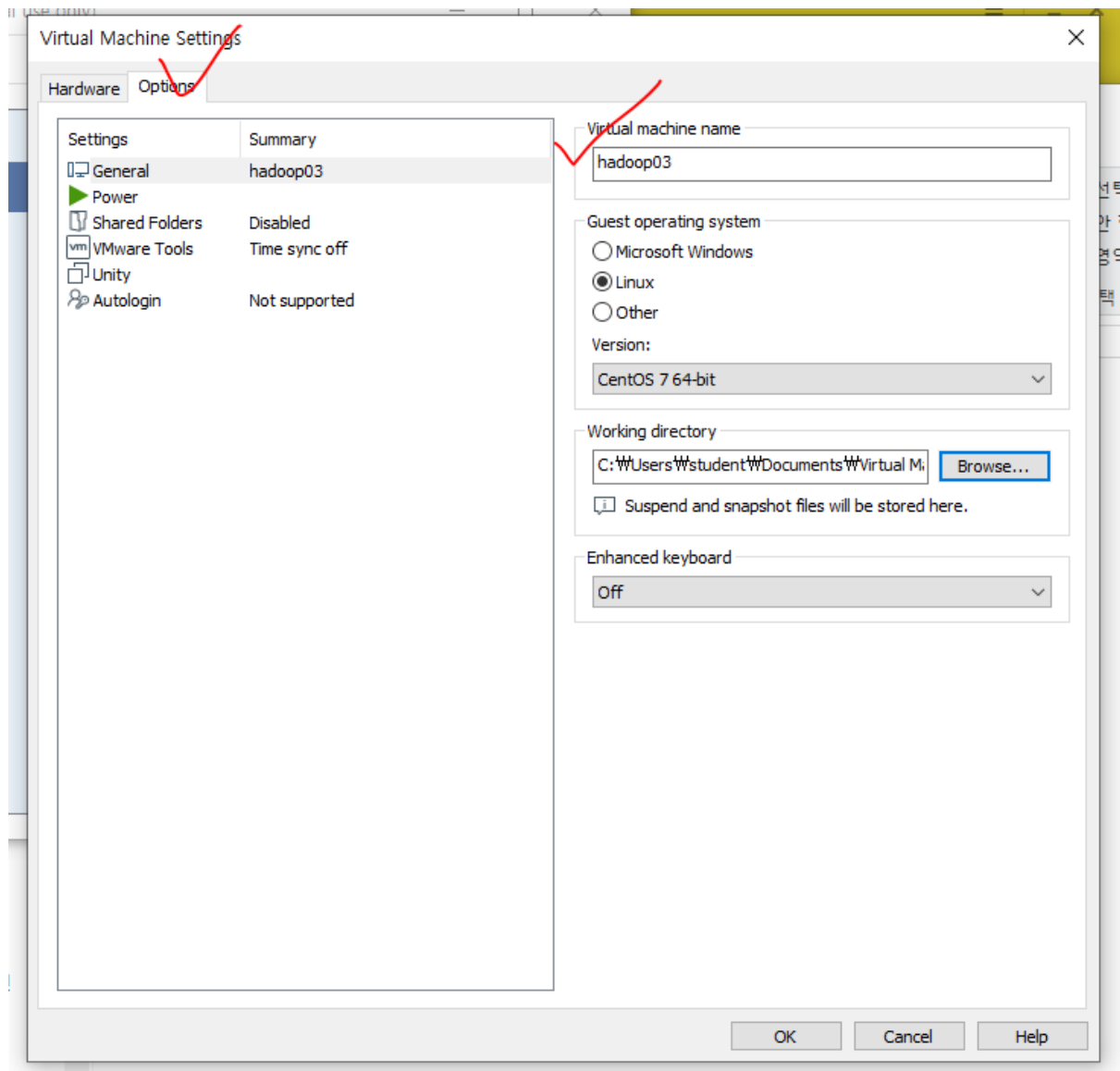
1. 머신 복제

- ip확인

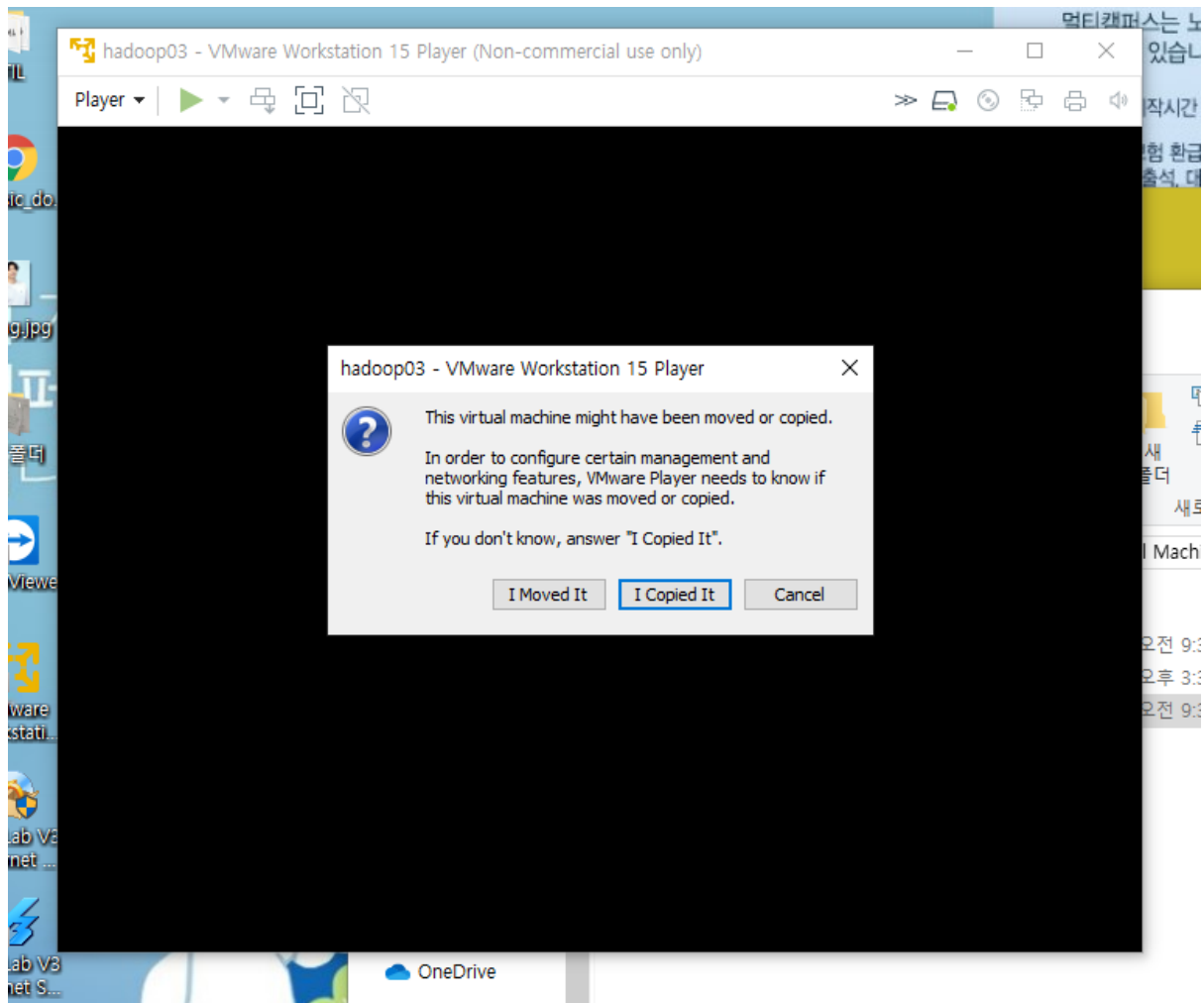
2. 머신 4대를 클러스터링

- 방화벽 해제
- hostname변경
- DNS설정
 - ▼ hosts파일 등록
 - ▼ 네트워크 프로세스를 restart
 - ▼ 설정확인 - 설정을 성공완료했는지 확인
 - ▼ 4대에 모두적용되도록 hadoop01머신에서 hadoop02, hadoop03, hadoop04에 직접접속

1. 머신 복제



이름이렇게바꿀수있다.



가상머신을 복사해서 붙여넣은 뒤 실행하면, 해당 창이 뜬다
 copy it 하면 새로 ip받음
 move it 하면 아이피 그대로 받음
 [I copied it 선택하자!]

- `ifconfig` 명령어를 치면 현재 머신의 ip주소를 알 수 있다.

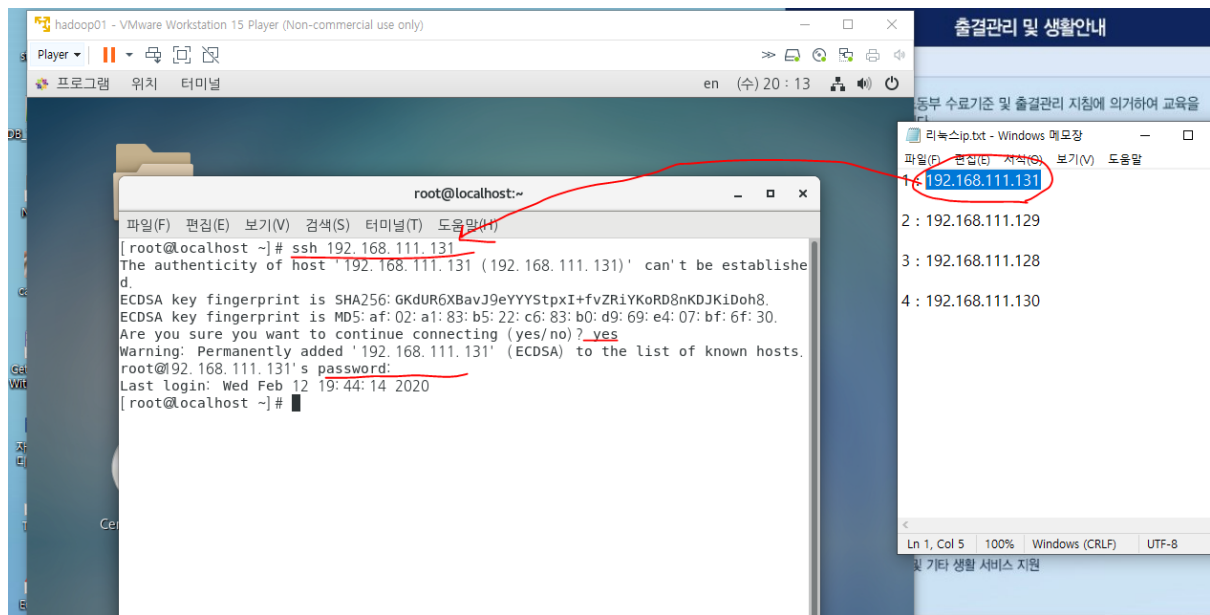
2. 클러스터링

머신 4개를 하나의 서버인 것처럼 사용

•

4개의 리눅스 서버를 열어서 접속되는지 확인해본다.

붙여넣기 : shift + insert



4개 다 들어가지는지 접속한 뒤, 내가 접속한 머신에서 exit 명령어를 통해 빠져나온다.

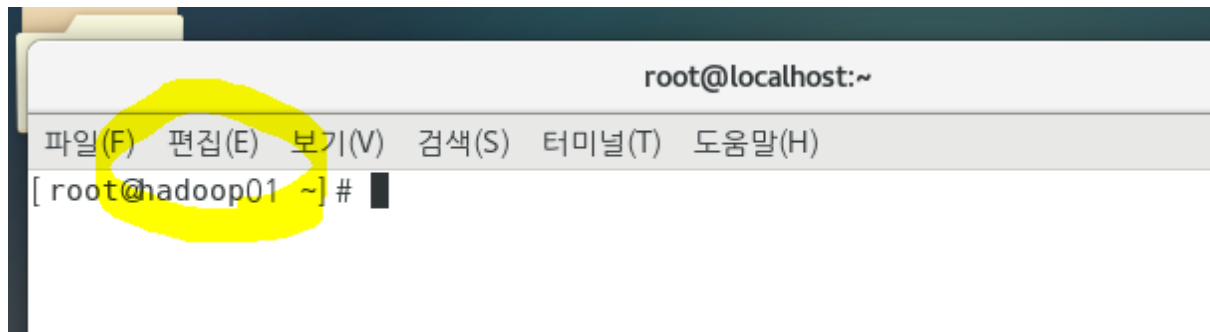
주로 작업할 공간은 hadoop1이며 다른 머신에서 작업하지 않도록 유의한다.

하지만 내가 어디에 접속했는지 잘 보이지 않는다. 내가 접속한 곳을 알아보려면 다음과 같은 명령어를 적어준다.

▼ hostnamectl set-hostname hadoop01

(첫 글자를 치고 tab을 몇 번 누르면 사용가능 명령어목록이 나온다.)

```
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@localhost ~]# hostnamectl set-hostname hadoop01
root@localhost ~]#
```



터미널창 나갔다 들어오면 변경됨을 확인할 수 있다.

다른위치 > 컴퓨터 > etc > hostname검색 > 오른쪽버튼눌러서 텍스트편집기로 열기 > 그러면 내가 저장한 이름이 적혀있다.

방화벽 해제하기

방화벽 확인하기

- `systemctl` : 서비스 상태 관련 명령어.

```
[root@hadoop01 ~] # systemctl list-units --type=service
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
abrt-ccpp.service	loaded	active	exited	Install ABRT coredump hook
abrt-oops.service	loaded	active	running	ABRT kernel log watcher
abrt-xorg.service	loaded	active	running	ABRT Xorg log watcher
abrttd.service	loaded	active	running	ABRT Automated Bug Reporting
accounts-daemon.service	loaded	active	running	Accounts Service
alsa-state.service	loaded	active	running	Manage Sound Card State (rest
atd.service	loaded	active	running	Job spooling tools
auditd.service	loaded	active	running	Security Auditing Service
avahi-daemon.service	loaded	active	running	Avahi mDNS/DNS-SD Stack

해당 명령어를 통해 서비스중인 것들 확인할 수 있다.

엔터를 쳐서 다음 목록을 읽을 수 있다. (끝까지 확인 후 컨트롤 c를 눌러 빠져나올 수 있다.)

여기서 `firewalld.service` 가 방화벽 부분이다.

방화벽은 디폴트로 설정되어있는 상태이다. 4대가 통신하기위해서 방화벽을 해제할 것이다.

`systemctl status firewalld` 명령어를 통해 방화벽 정보를 볼 수 있다.

```
[root@hadoop01 ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since 수 2020-02-12 19:43:40 KST; 2h 58min ago
     Docs: man:firewalld(1)
  Main PID: 1135 (firewalld)
    Tasks: 2
   CGroup: /system.slice/firewalld.service
           └─1135 /usr/bin/python2 -Es /usr/sbin/firewalld --nofork --nopid

2월 12 19:43:39 localhost.localdomain systemd[1]: Starting firewalld - dyna...
2월 12 19:43:40 localhost.localdomain systemd[1]: Started firewalld - dynam...
Hint: Some lines were ellipsized, use -l to show in full.
[root@hadoop01 ~]#
```

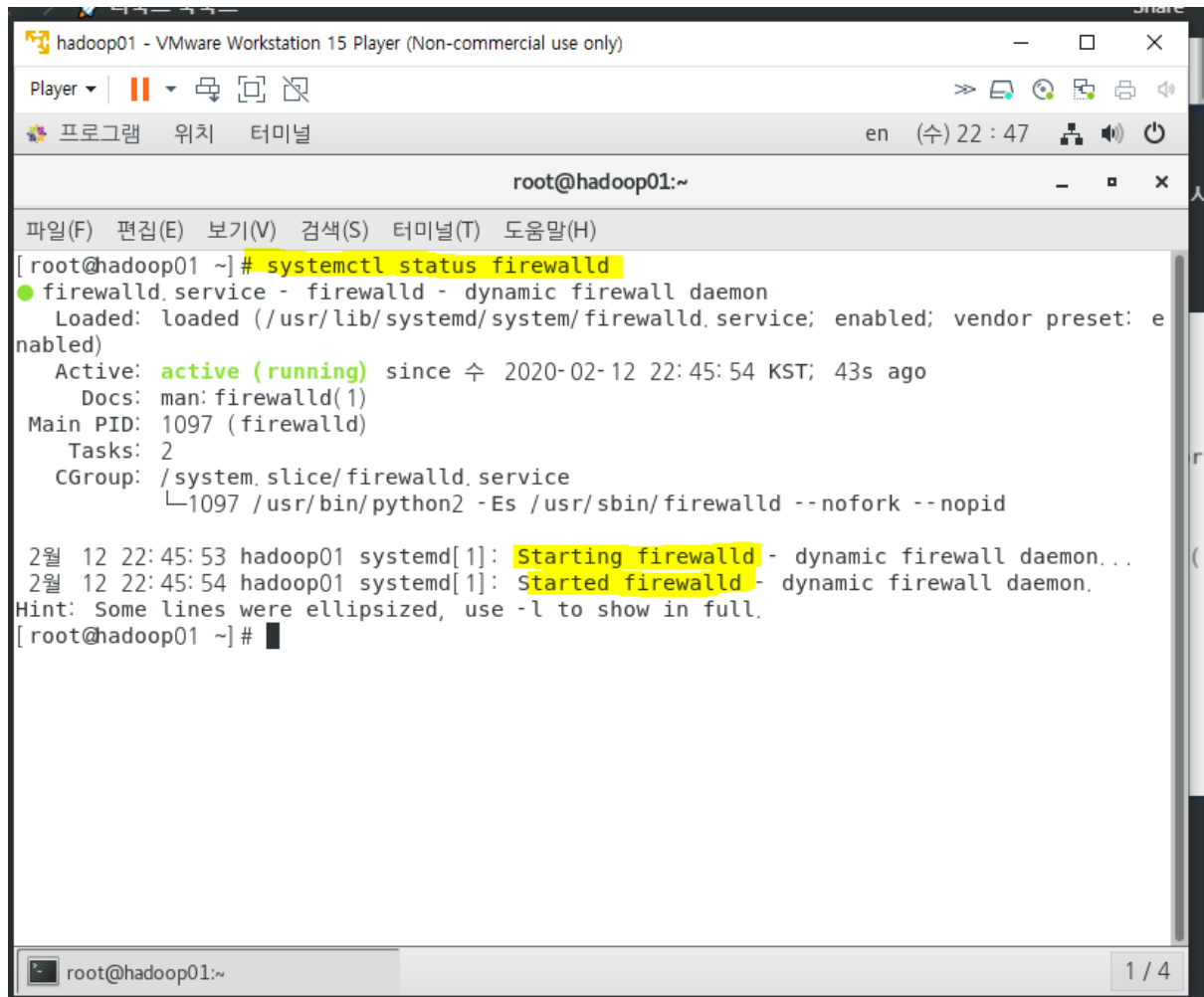
start 되어있는 상태이므로 서비스를 멈추기 위해 `systemctl stop firewalld` 명령어를 통해 중지시킨다.

```
[root@hadoop01 ~]# systemctl stop firewalld
[root@hadoop01 ~]# systemctl status firewalld
Unknown operation 'status'.
[root@hadoop01 ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since 수 2020-02-12 22:44:17 KST; 17s ago
     Docs: man:firewalld(1)
  Process: 1135 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=exited, status=0/SUCCESS)
 Main PID: 1135 (code=exited, status=0/SUCCESS)

2월 12 19:43:39 localhost.localdomain systemd[1]: Starting firewalld - dyna...
2월 12 19:43:40 localhost.localdomain systemd[1]: Started firewalld - dynam...
2월 12 22:44:17 hadoop01 systemd[1]: Stopping firewalld - dynamic firewall...
2월 12 22:44:17 hadoop01 systemd[1]: Stopped firewalld - dynamic firewall...n.
Hint: Some lines were ellipsized, use -l to show in full.
[root@hadoop01 ~]#
```

그리고 나서 다시 정보를 보면 stop됨을 확인할 수 있다

이번에는 `reboot` 명령어를 쳐서 다시시작을 시키고, 다시 상태확인 명령어를 통해 확인해보면



```
hadoop01 - VMware Workstation 15 Player (Non-commercial use only)
Player | [Icons] | en (수) 22 : 47
프로그램 위치 터미널
root@hadoop01:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@hadoop01 ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since 수 2020-02-12 22:45:54 KST; 43s ago
     Docs: man:firewalld(1)
    Main PID: 1097 (firewalld)
       Tasks: 2
      CGroup: /system.slice/firewalld.service
              └─1097 /usr/bin/python2 -Es /usr/sbin/firewalld --nofork --nopid

2월 12 22:45:53 hadoop01 systemd[1]: Starting firewalld - dynamic firewall daemon...
2월 12 22:45:54 hadoop01 systemd[1]: Started firewalld - dynamic firewall daemon.
Hint: Some lines were ellipsized, use -l to show in full.
[root@hadoop01 ~]#
```

방화벽이 다시 start 상태로 된 것을 확인할 수 있다.

다시시작을 하면 방화벽은 다시 디폴트상태(start)로 돌아온다.

따라서 `systemctl disabled firewalld` 명령어를 통해 재부팅 했을 경우 방화벽 사용중지 상태를 유지하도록 한다.

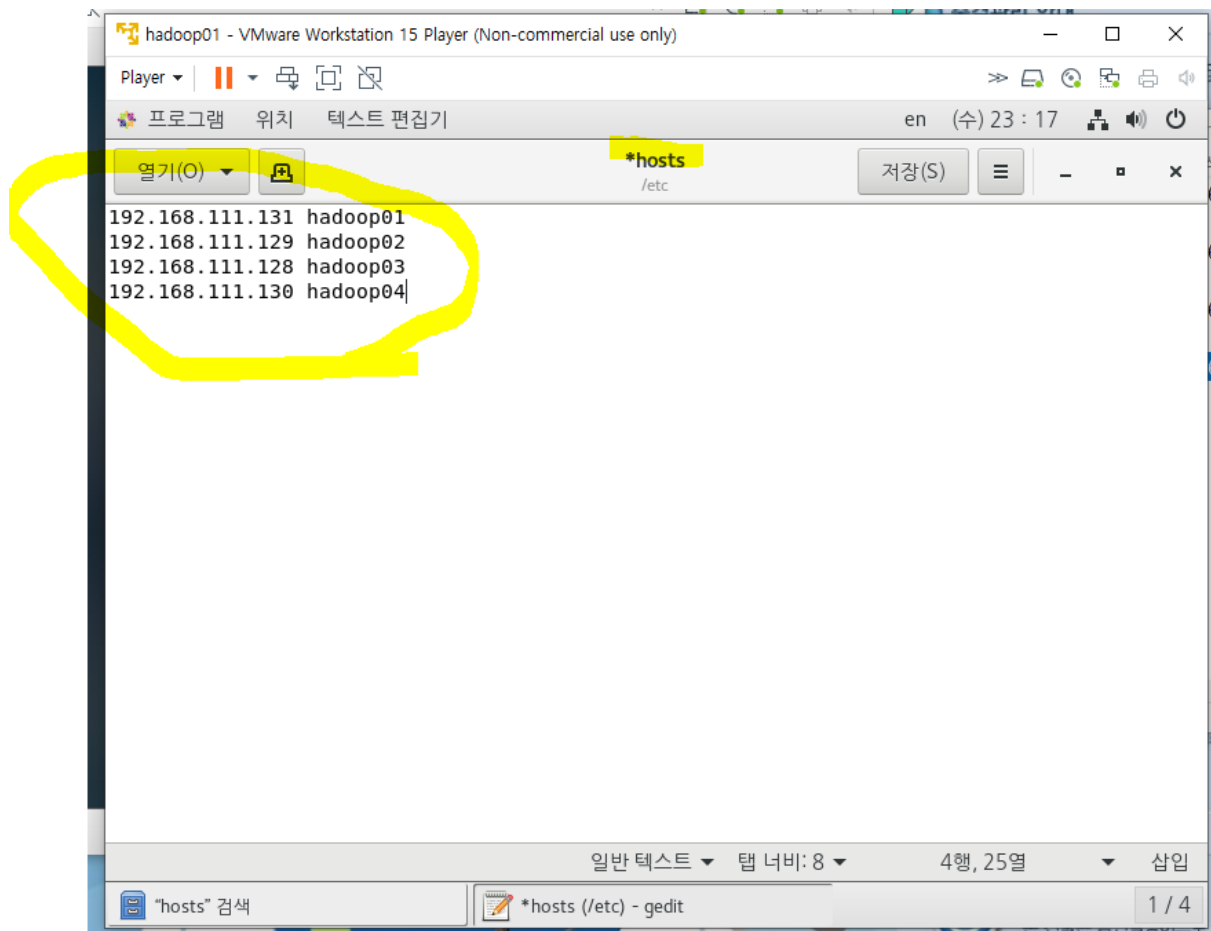
```
[root@hadoop01 ~]# systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@hadoop01 ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset:
   enabled)
   Active: active (running) since 수 2020-02-12 22:45:54 KST; 4min 19s ago
     Docs: man:firewalld(1)
    Main PID: 1097 (firewalld)
    CGroup: /system.slice/firewalld.service
            └─1097 /usr/bin/python2 -Es /usr/sbin/firewalld --nofork --nopid

2월 12 22:45:53 hadoop01 systemd[1]: Starting firewalld - dynamic firewall daemon...
2월 12 22:45:54 hadoop01 systemd[1]: Started firewalld - dynamic firewall daemon.
Hint: Some lines were ellipsized, use -l to show in full.
[root@hadoop01 ~]#
```

DNS 설정하기(hostname 변경)

기존에 네트워크 접속 할 때, `ssh ip` 로 접속을 했는데 ip를 다 쓰는 것보다 머신의 이름으로 접속하는 것이 편하다.

- ▼ 다른위치 > 컴퓨터 > etc > hosts 검색 후 해당 파일 텍스트편집기로 열기 > 적혀있는 모든 내용을 지운다.
- ▼ `ip 머신명` 형태로 적어준다.
- ▼ ip주소 작성 시, 머신마다 ifconfig 명령어를 통해 ip주소를 반드시 확인하자

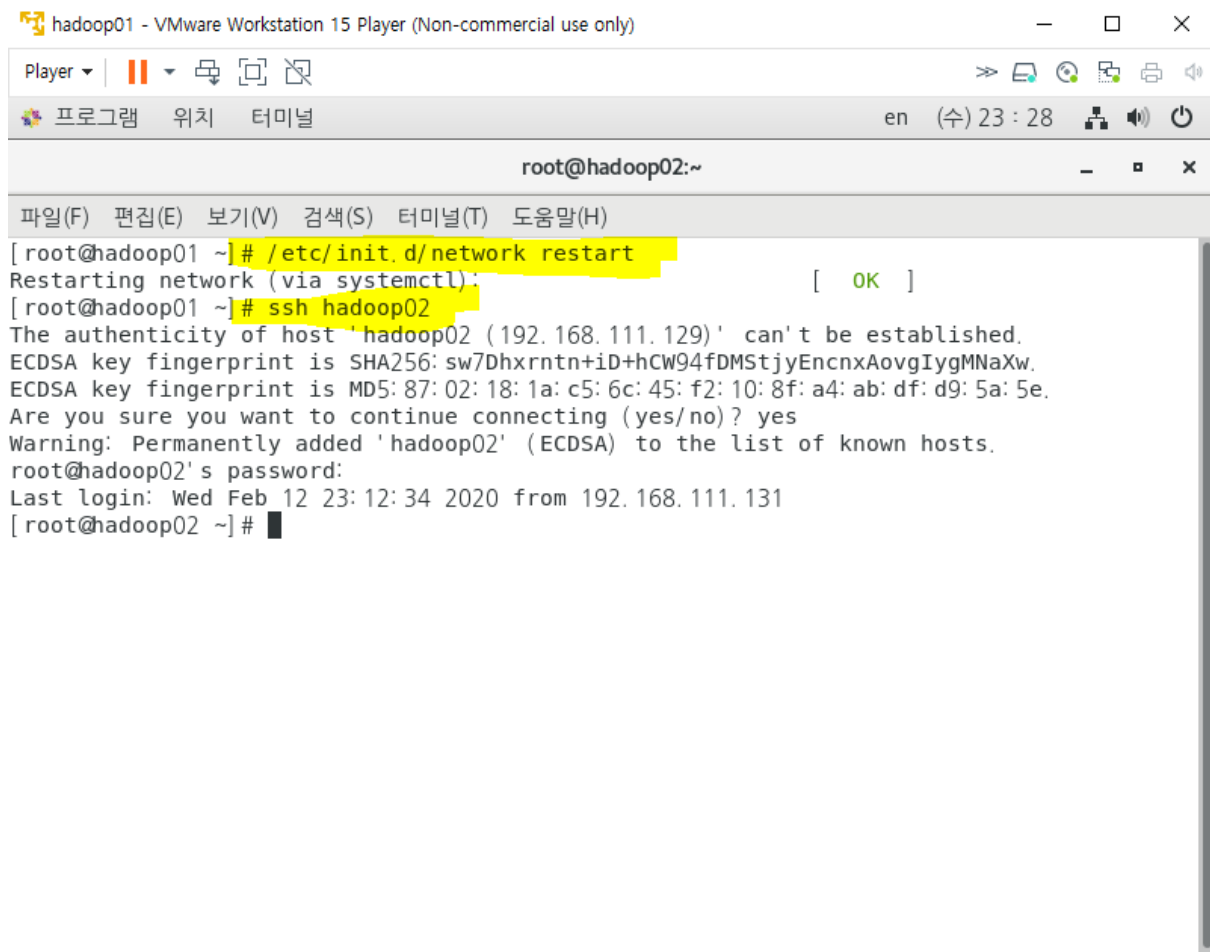


네트워크 프로세스 restart

네트워크 설정정보는 etc에 있다.

`/etc/init.d/network restart` 명령어를 실행해 restart 시킨다.

그리고 `ssh 머신명` 으로 접속해보자.



```
hadoop01 - VMware Workstation 15 Player (Non-commercial use only)
Player ▾ | || ▢ ▢ ▢
프로그램 위치 터미널 en (수) 23 : 28
root@hadoop02:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[ root@hadoop01 ~]# /etc/init.d/network restart
Restarting network (via systemctl): [ OK ]
[ root@hadoop01 ~]# ssh hadoop02
The authenticity of host 'hadoop02 (192.168.111.129)' can't be established.
ECDSA key fingerprint is SHA256:sw7Dhxrntn+iD+hCW94fDMStjyEncnxAovgIygMNaXw.
ECDSA key fingerprint is MD5:87:02:18:1a:c5:6c:45:f2:10:8f:a4:ab:df:d9:5a:5e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop02' (ECDSA) to the list of known hosts.
root@hadoop02's password:
Last login: Wed Feb 12 23:12:34 2020 from 192.168.111.131
[ root@hadoop02 ~]#
```

이를 통해 ip가 아닌 도메인을 통해 접속이 가능함을 알 수 있다.

확인 후 `exit` 명령어로 접속을 해제하고, 다른 머신들도 이름으로 접근 가능한지 확인한다.

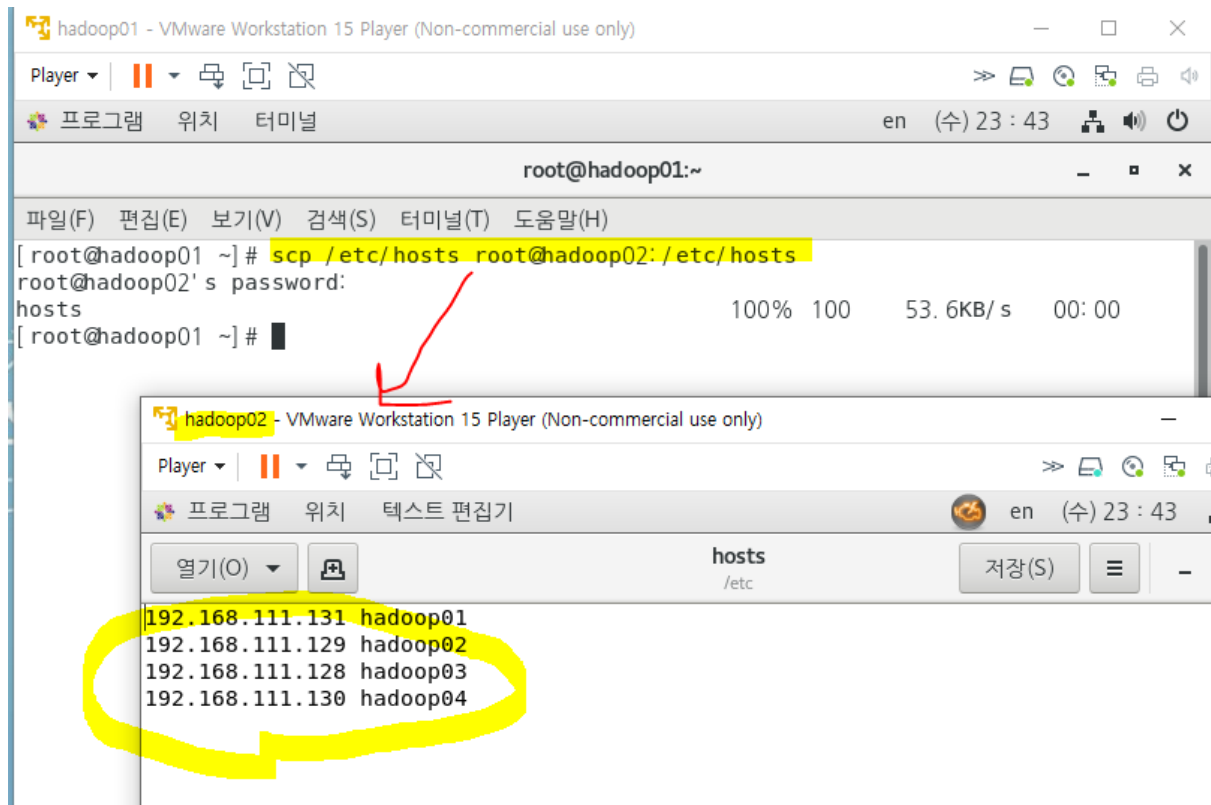
[원격 서버로 copy]

`scp` copy할파일(위치까지명시) copy받을서버의위치

`scp /etc/hosts root@hadoop02:/etc/hosts`

===

↳>명령어 ↳>copy할 파일 ↳> target서버의 위치와 파일명



hadoop01의 hosts파일을 hadoop2에 copy한다. 이제 hadoop2도 네이밍이 등록되었다.

hadoop02, 03, 04모두 copy해준다. 설정파일이 바뀌었기 때문에 1과 마찬가지로 모두 `/etc/init.d/network restart` 명령어를 실행해 restart 시켜줘야 한다.

[원격 서버에 실행명령]

ssh 서버 "실행할명령문"

==> 도메인

암호화된 통신을 위해서 공개키 생성 후 배포

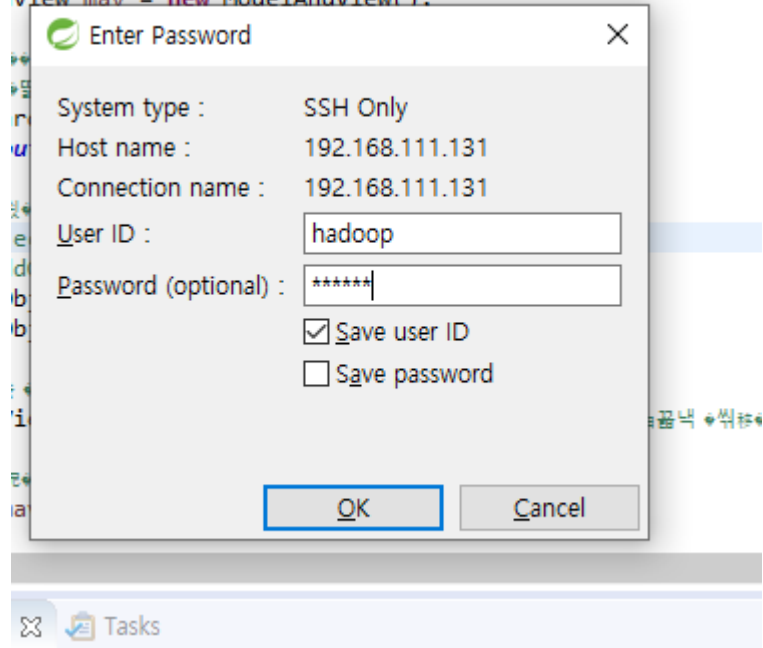
하둡실행을 하둡계정에서 한다.

하둡 홈디렉토리의 위치는 ? 내폴더 > 다른위치 > 컴퓨터 > home

스프링에서 id, pass 를 hadoop, hadoop으로 하여 hadoop으로 접속

ip주소 오른쪽버튼 > connect >

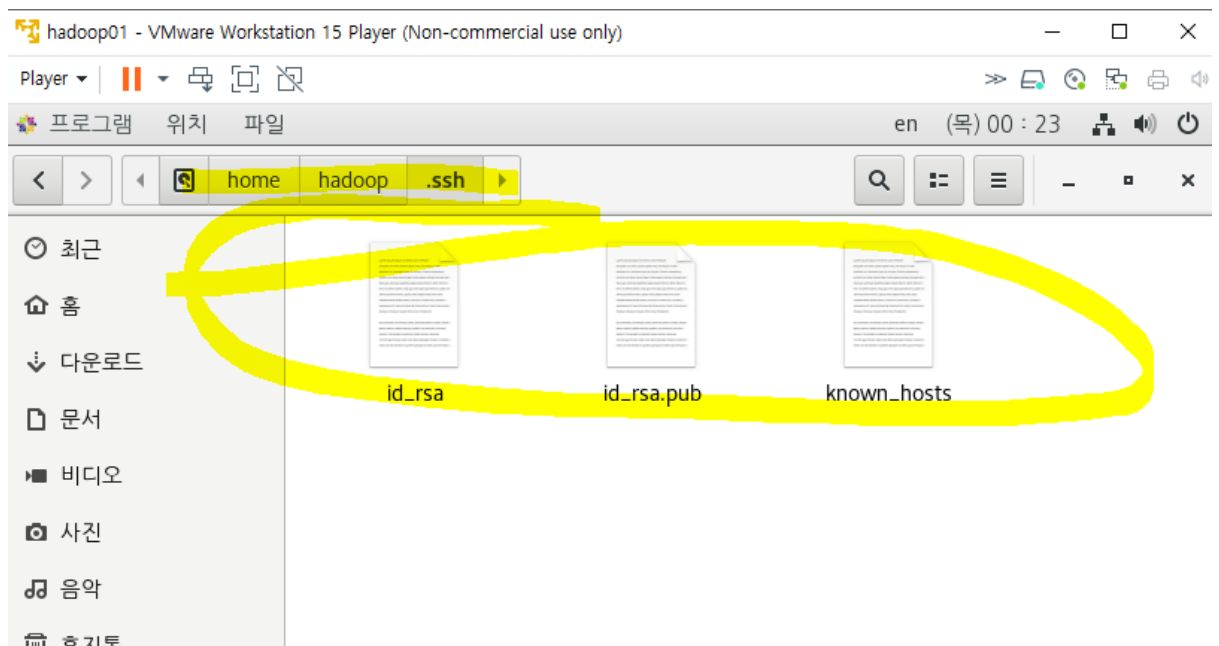
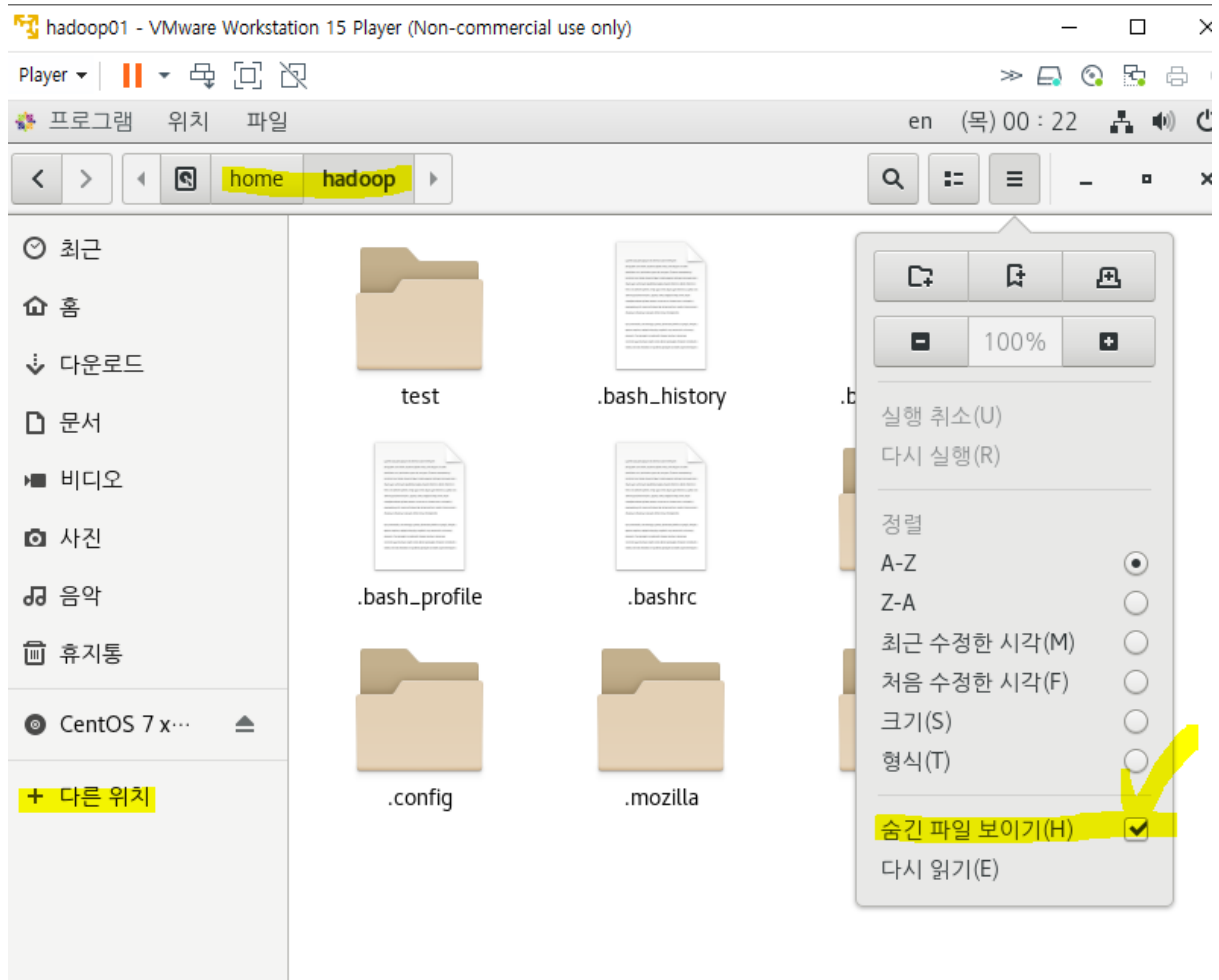
```
ut.println( category=> +category)); //변경사항  
View mav = new ModelAndView();
```

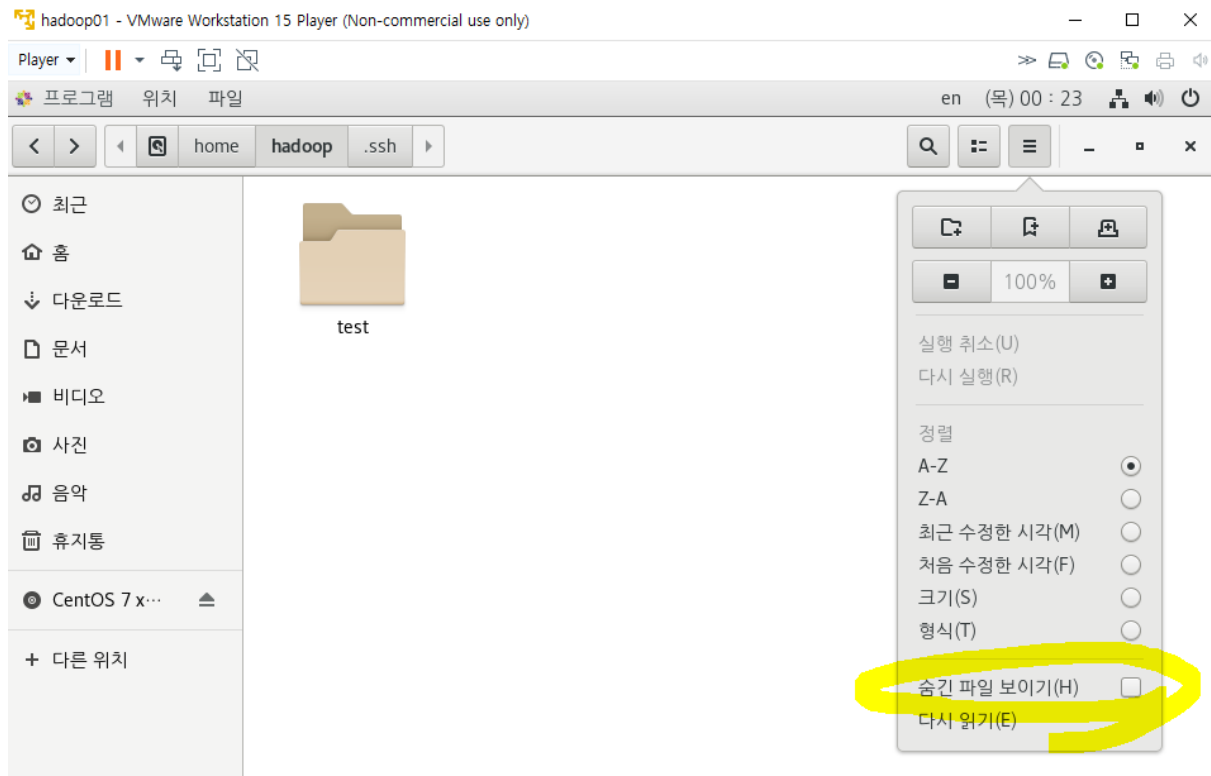


```
[root@hadoop01 ~]# su hadoop  
[hadoop@hadoop01 root]$ cd ~  
[hadoop@hadoop01 ~]$ ssh hadoop02  
The authenticity of host 'hadoop02 (192.168.111.129)' can't be established.  
ECDSA key fingerprint is SHA256: sw7Dhxrntn+id+hCW94fDMStjyEncnxAovgIygMNaXw.  
ECDSA key fingerprint is MD5: 87: 02: 18: 1a: c5: 6c: 45: f2: 10: 8f: a4: ab: df: d9: 5a: 5e.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'hadoop02,192.168.111.129' (ECDSA) to the list of known hosts.  
hadoop@hadoop02's password:  
[hadoop@hadoop02 ~]$ exit
```

0. 키 생성 (RSA)

```
[hadoop@hadoop01 ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:VLeh2rt5Hi0FGgBnIrrgssSlq5mqPZVGebTofMFnXX0 hadoop@hadoop01
The key's randomart image is:
+---[RSA 2048]-----+
|      . o. + . o.      |
|      . .. + o o. o. E|
|      . = . . . o. o .  |
| o. o. + = . oo. o .   |
| . = + o +S o .        |
| o.. * . . . o        |
| .. o . . . o .       |
| . + . . . o. o       |
| 0 .. . oo.          |
+---[SHA256]-----+
[hadoop@hadoop01 ~]$
```





숨긴파일보이기 해제하면 다시 안보인다.

공개키는 2,3,4번이 갖고있고 비공개키는 1번이 갖고있다.

2번에서3번으로 갈 때는 비밀번호 물어본다. (2번은 3번에대한 비공개키를 갖고있지 않기 때문) 마스터에서 슬레이브들에 접속할 때 비밀번호 안물어보게 하기 위해서 해주는 작업이다.

`id_rsa. pub` 가 공개키이다.

먼저 root에서 hadoop01로 간뒤 공개키를 copy하는 작업을 한다.

- ① `su hadoop`
- ② `cd ~`
- ③ `cd.ssh`
- ④ `ls`
- ⑤ `ssh-copy-id -i id_rsa.pub hadoop@hadoop02`

- ⑥ `ssh hadoop02` 로 hadoop2에 접근하면 비밀번호 입력 없이 접근 가능하다.

공개키를 hadoop02 로 카피

```
ssh-copy-id -i id_rsa.pub hadoop@hadoop02
```

```
[root@hadoop01 ~]# su hadoop
[hadoop@hadoop01 root]$ cd~
bash: cd~: 명령을 찾을 수 없습니다...
[hadoop@hadoop01 root]$ cd ~
[hadoop@hadoop01 ~]$ cd .ssh
[hadoop@hadoop01 .ssh]$ ls
id_rsa id_rsa.pub known_hosts
[hadoop@hadoop01 .ssh]$ ssh-copy-id -i id_rsa.pub hadoop@hadoop02
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
hadoop@hadoop02's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'hadoop@hadoop02'"
and check to make sure that only the key(s) you wanted were added.
[hadoop@hadoop01 .ssh]$
```

공개키를 보냈기 때문에, hadoop01에서 다른계정으로 접근할 때 더 이상 비밀번호를 입력하지 않아도 된다.

```
Now try logging into the machine, with: "ssh 'hadoop@hadoop02'"
and check to make sure that only the key(s) you wanted were added.

[hadoop@hadoop01 .ssh]$ ssh hadoop02
Last login: Thu Feb 13 00:19:46 2020 from hadoop01
[hadoop@hadoop02 ~]$
```

같은 작업을 hadoop03, hadoop04 에도 해준다.

공개키는 2,3,4번이 갖고있고 비공개키는 1번이 갖고있다.

2번에서3번으로 갈 때는 비밀번호 물어본다.

마스터에서 슬레이브들에 접속할 때 비밀번호 안물어보게 하기 위해서 해준 작업이다.

네트워크 구성

[hostonly]

- 호스트 내 게스트들끼리 소통/가상머신들끼리소통

[nat]

- 공유기로 접근해서 인터넷으로 나감

