
Apache Hadoop install guide

1. 구성 환경 및 사전 준비

호스트OS Windows 10 HOME

로컬PC가 공유기(WI-FI)를 통해 인터넷이 연결되어 있다는 가정

하둡 클러스터에 대한 이해와 테스트를 목적으로 하는 입문자용 가이드

- jdk 압축 파일 (본 가이드에서는 jdk-8u191-linux-x64.tar)
 - <https://www.oracle.com/kr/java/technologies/javase/javase8-archive-downloads.html>
- CentOS7 디스크 이미지 파일
 - http://isoredirect.centos.org/centos/7/isos/x86_64/ (* 최상단 kakao 혹은 naver가 빠름)
- virtualbox 6.1
 - <https://www.virtualbox.org/wiki/Downloads> (* Windows hosts 클릭, 다운로드 후 .exe 실행)
- MobaXterm 8.6
 - <https://mobaxterm.software.informer.com/8.6/> (* 다운로드 후 .exe 실행)



2. 가상 서버 생성

윈도우 탐색기 – cmd – ipconfig 입력 – VirtualBox Host-Only Network 주소 확인

명령 프롬프트

```
C:\Users\KWC>ipconfig
```

Windows IP 구성

이더넷 어댑터 VirtualBox Host-Only Network:

```
연결별 DNS 접미사. . . . . :  
링크-로컬 IPv6 주소. . . . . : fe80::b00c:47ed:3edb:eed9%37  
IPv4 주소. . . . . : 192.168.56.1  
서브넷 마스크. . . . . : 255.255.255.0  
기본 게이트웨이. . . . . :
```

무선 LAN 어댑터 로컬 영역 연결* 1:

```
미디어 상태. . . . . : 미디어 연결 끊김  
연결별 DNS 접미사. . . . . :
```

무선 LAN 어댑터 로컬 영역 연결* 2:

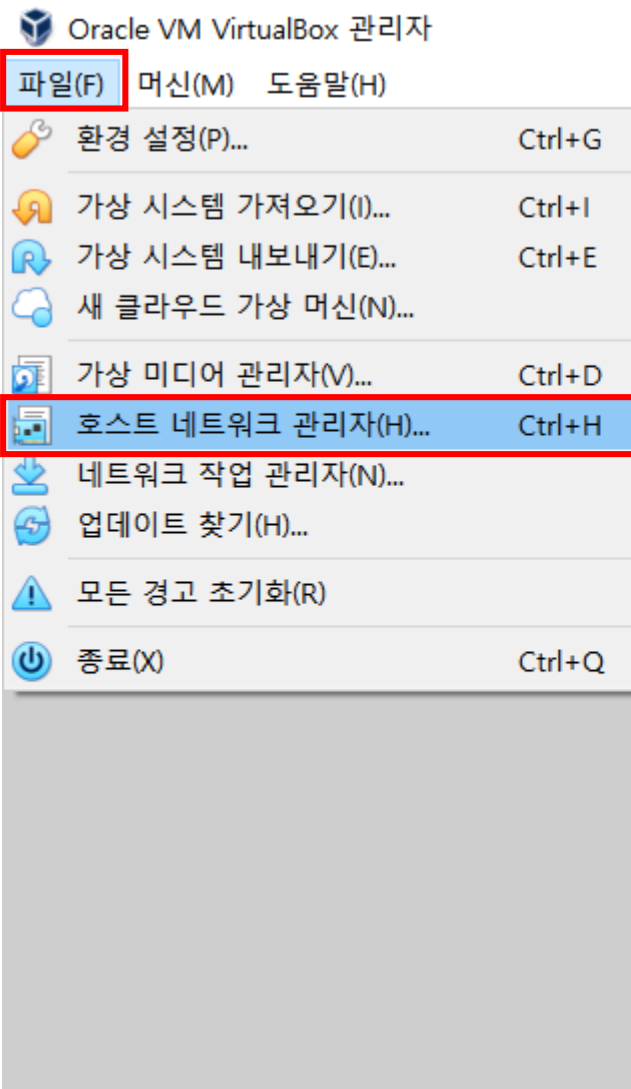
```
미디어 상태. . . . . : 미디어 연결 끊김  
연결별 DNS 접미사. . . . . :
```

무선 LAN 어댑터 Wi-Fi:

```
연결별 DNS 접미사. . . . . :  
링크-로컬 IPv6 주소. . . . . : fe80::e03a:2a92:308a:4020%3  
IPv4 주소. . . . . : 10.10.11.71  
서브넷 마스크. . . . . : 255.255.255.0  
기본 게이트웨이. . . . . : 10.10.11.1
```

2. 가상 서버 생성

Virtual box 실행 – 좌측 상단 파일 – 호스트 네트워크 관리자 – 아래와 같이 설정 - 적용



환경 설정(P) 가져오기 내보내기 새로 만들기(N) 추가(A)

VirtualBox에 오신 것을 환영합니다!

이 프로그램의 왼쪽 부분은 전역 도구 및 컴퓨터에 있는 모든 가상 머신과 가상 머신 그룹 목록을 표시합니다. 도구 모음의 단추를 사용하여 새로운 가상 머신을 만들거나, 추가하거나, 가져올 수 있습니다. 현재 선택한 구성 요소에 사용할 수 있는 도구 모음 단추를 눌러 해당하는 도구 모음을 호출할 수 있습니다.

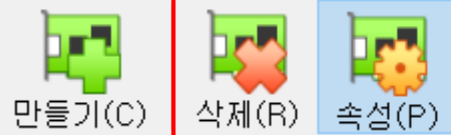
F1 키를 누르면 상황에 맞는 도움말을 볼 수 있으며, 최근 정보와 뉴스를 보려면 www.virtualbox.org를 방문하십시오.



2. 가상 서버 생성

호스트 네트워크 관리자

네트워크(N)



기존에 어댑터가 없을 시
만들기 버튼으로 생성

이름	IPv4 주소/마스크	IPv6 주소/마스크	DHCP 서버
VirtualBox Host-Only Ethernet Adapter	192.168.56.1/24		<input type="checkbox"/> 사용함

어댑터(A) DHCP 서버(D)

- ☐ 자동으로 어댑터 설정(A)
☒ 수동으로 어댑터 설정(M)

IPv4 주소(I): 192.168.56.1

IPv4 서브넷 마스크(M): 255.255.255.0

IPv6 주소(P): fe80::b00c:47ed:3edb:eed9

IPv6 접두사 길이(L): 64

초기화

적용

닫기

2. 가상 서버 생성

새로 만들기 - 이름 설정 - 다음

(이름 설정에 'cent'를 입력하면 자동으로 리눅스 계열로 설정된다, 후 원하는 서버 이름 입력)

Oracle VM VirtualBox 관리자

파일(F) 머신(M) 도움말(H)

가상 머신 만들기

이름 및 운영 체제

새 가상 머신을 나타내는 이름과 저장할 대상 폴더를 입력하고 설치할 운영 체제를 선택하십시오. 입력한 이름은 VirtualBox에서 가상 머신을 식별하는 데 사용됩니다.

이름:

머신 폴더:

종류(T):

버전(V):

전문가 모드(E)

다음(N)

취소


새로 만들기(N)

추가(A)

환영합니다!

전역 도구 및 컴퓨터에 있는 모든 목록을 표시합니다. 도구 모음의 가상 머신을 만들거나, 추가하거나, 선택한 구성 요소에 사용할 수 있는 하는 도구 모음을 호출할 수 있습니다.

도움말을 볼 수 있으며, 최근 정 [ualbox.org](http://www.virtualbox.org)를 방문하십시오.



2. 가상 서버 생성

메모리 크기 설정 - 다음

? ×

← 가상 머신 만들기

메모리 크기

가상 머신에 할당할 메모리(RAM) 크기를 메가바이트 단위로 입력하십시오.

추천 메모리 크기는 **1024** MB입니다.

4 MB

32768 MB

2048 MB

다음(N)

취소

2. 가상 서버 생성

만들기

? ×

← 가상 머신 만들기

하드 디스크

필요하다면 새 가상 머신에 가상 하드 디스크를 추가할 수 있습니다. 새 하드 디스크 파일을 만들거나, 목록에서 선택하거나, 폴더 아이콘을 통하여 다른 위치에 있는 가상 하드 디스크 파일을 선택할 수 있습니다.

더 자세한 구성이 필요하다면 이 단계를 건너뛰고 가상 머신을 만든 다음 설정을 진행하십시오.

추천하는 하드 디스크 크기는 **8.00 GB**입니다.

- ☐ 가상 하드 디스크를 추가하지 않음(D)
- ☒ 지금 새 가상 하드 디스크 만들기(C)
- ☐ 기존 가상 하드 디스크 파일 사용(U)

비어 있음

만들기

취소

2. 가상 서버 생성

다음

? X

← 가상 하드 디스크 만들기

하드 디스크 파일 종류

새 가상 하드 디스크 파일 형식을 선택하십시오. 다른 가상화 소프트웨어에서 디스크를 사용하지 않으려면 선택을 변경하지 않아도 됩니다.

☒ VDI(VirtualBox 디스크 이미지)

☐ VHD(가상 하드 디스크)

☐ VMDK(가상 머신 디스크)

전문가 모드(E)

다음(N)

취소

2. 가상 서버 생성

선택 후 다음

? ×

← 가상 하드 디스크 만들기

물리적 하드 드라이브에 저장

새 가상 하드 디스크 파일을 사용하는 대로 커지게 할 것인지(동적 할당) 최대 크기로 만들 것인지(정적 할당) 선택하십시오.

동적 할당 하드 디스크 파일은 가상 디스크를 사용할 때 **고정된 최대 크기까지** 파일 크기가 커지지만, 사용량이 줄어들어도 자동적으로 작아지지는 않습니다.

고정 크기 하드 디스크 파일은 만드는 데 더 오래 걸리지만 사용할 때 더 빠릅니다.

☐ 동적 할당(D)

☒ 고정 크기(F)

다음(N)

취소

2. 가상 서버 생성


디스크 크기 설정 후 만들기

? X

← 가상 하드 디스크 만들기

파일 위치 및 크기

새 가상 하드 디스크 파일의 이름을 아래 상자에 입력하거나 폴더 아이콘을 클릭해서 파일을 생성할 폴더를 지정할 수 있습니다.

C:\Users\WKC\VirtualBox VMs\server01\server01.vdi 

새 가상 하드 디스크 크기를 메가바이트 단위로 입력하십시오. 가상 머신에서 가상 하드 드라이브에 저장할 수 있는 데이터의 최대 크기입니다.

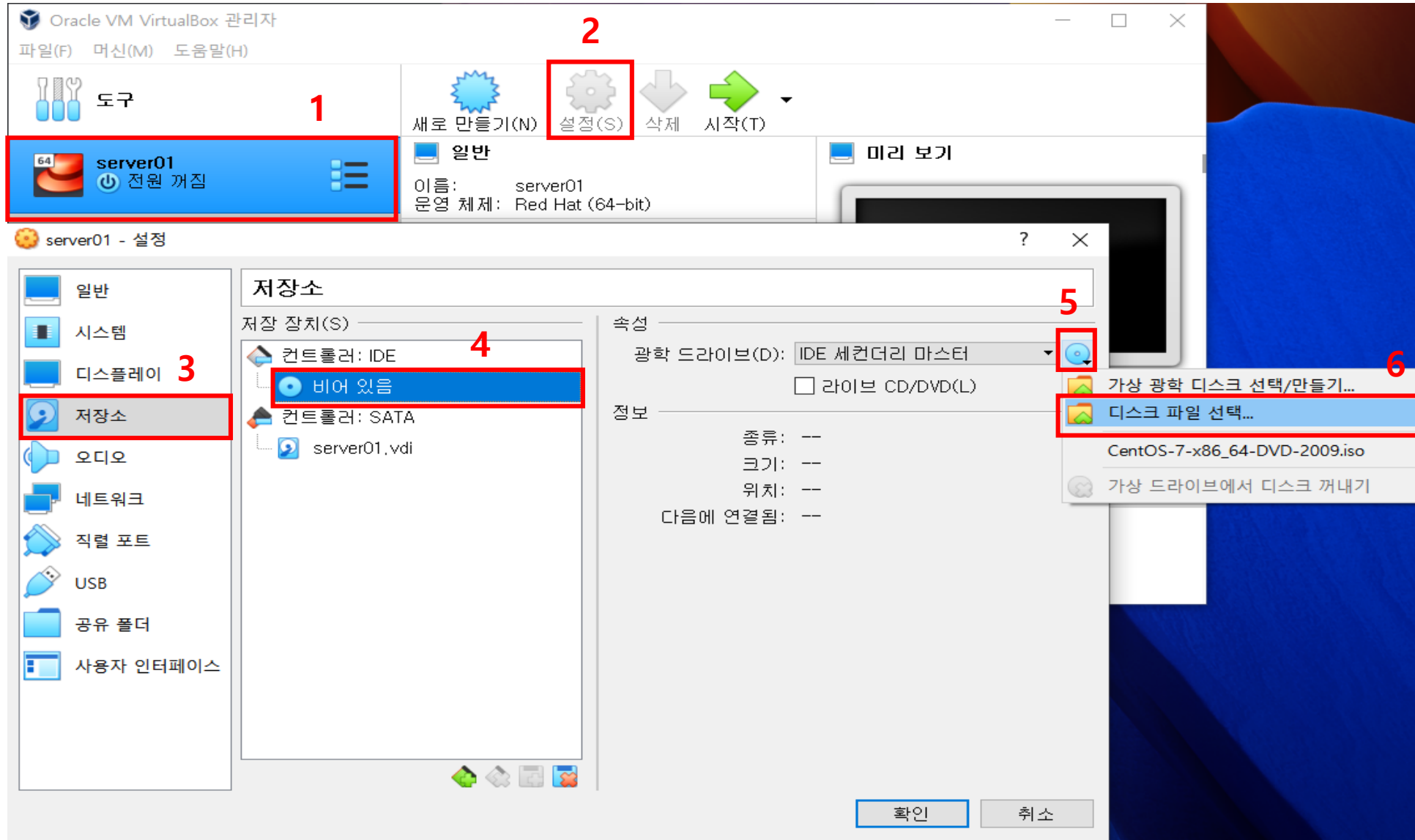


만들기

취소

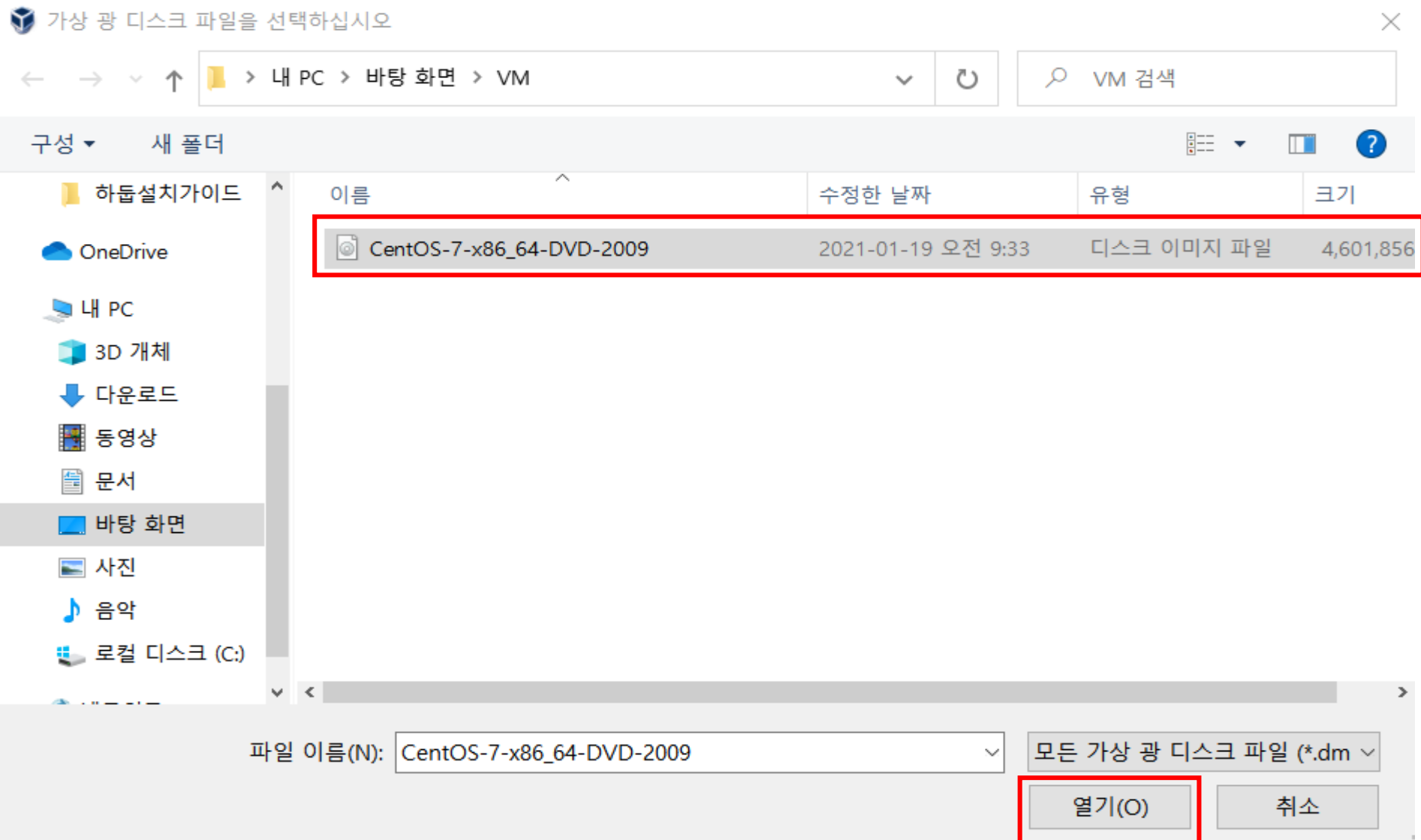
2. 가상 서버 생성

아래 순서대로 클릭



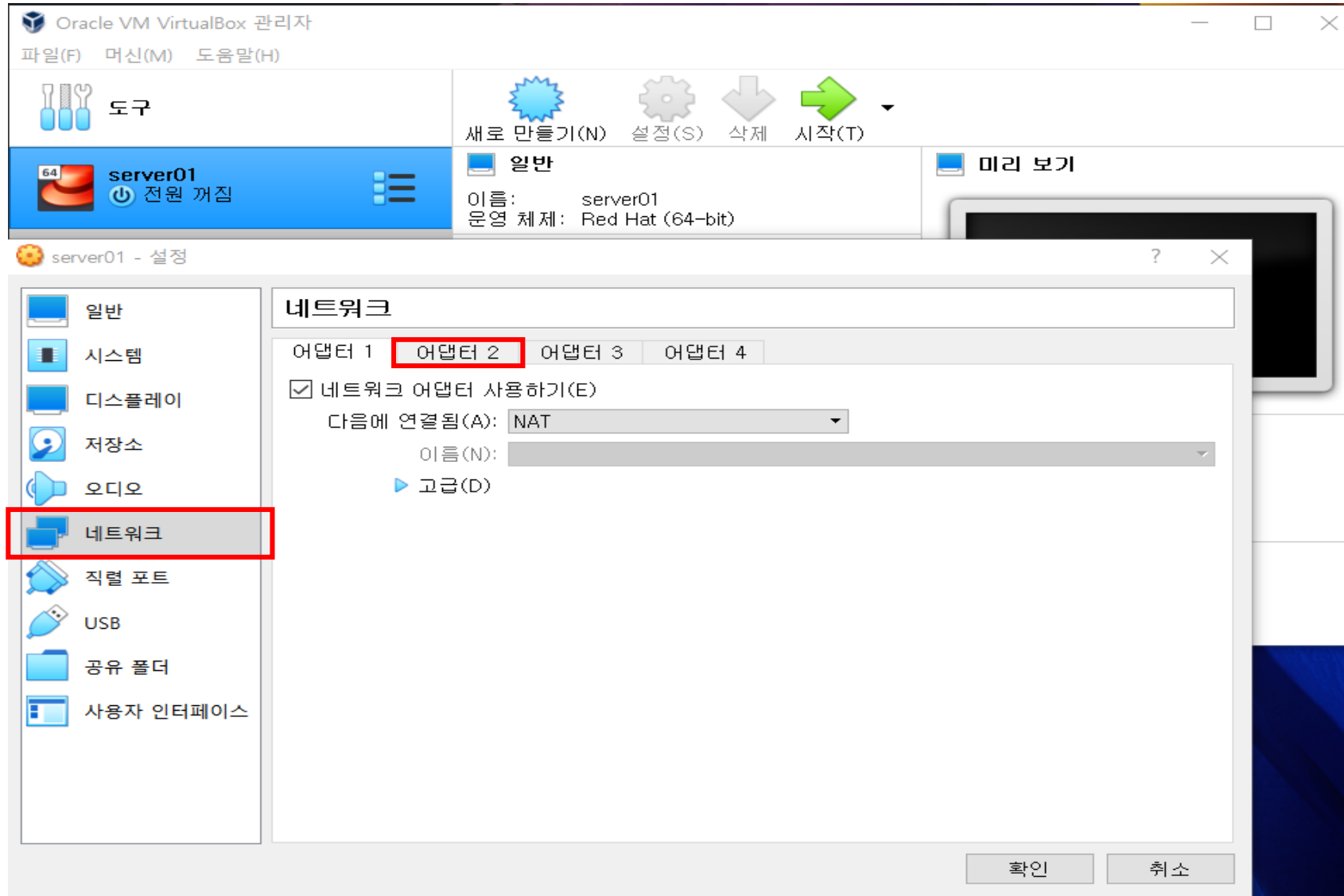
2. 가상 서버 생성

다운로드 받았던 디스크 이미지 파일 등록



2. 가상 서버 생성

네트워크 - 어댑터1 기본으로 설정 후 어댑터2 클릭



2. 가상 서버 생성

다음과 같이 설정 후 확인

server01 - 설정

네트워크

어댑터 1 어댑터 2 어댑터 3 어댑터 4

☒ 네트워크 어댑터 사용하기(E)

다음에 연결됨(A): 호스트 전용 어댑터

이름(N): VirtualBox Host-Only Ethernet Adapter

고급(D)

어댑터 종류(T): Intel PRO/1000 MT Desktop(82540EM)

무작위 모드(P): 모두 허용

MAC 주소(M): 080027731D18

☒ 케이블 연결됨(C)

포트 포워딩(P)

확인 취소

2. 가상 서버 생성

Virtual box 네트워크 어댑터 설정 방법:

가상 서버를 생성하였을 때, 기본적으로 어댑터 1에는 NAT 어댑터가 등록된다. NAT 어댑터는 게스트 OS(새로 생성한 가상서버)가 호스트OS(로컬PC의 운영체제 여기서는 WINDOWS 10)의 IP를 통해 외부 인터넷에 연결해주는 가장 간단한 방법이다. 하지만 게스트OS간의 통신이 불가능하다.

호스트 전용 어댑터는 외부 인터넷과는 단절되어 있지만 구축한 게스트OS간의 통신이 가능하다.

이 두 어댑터를 사용하여 서로의 단점을 보완, 게스트->외부 인터넷, 게스트 -> 게스트 통신이 가능하다.

[참조] <https://technote.kr/213>

2. 가상 서버 생성

시작

Oracle VM VirtualBox 관리자

파일(F) 머신(M) 도움말(H)

도구

새로 만들기(N) 설정(S) 삭제 시작(T)

server01
전원 꺼짐

일반
이름: server01
운영 체제: Red Hat (64-bit)

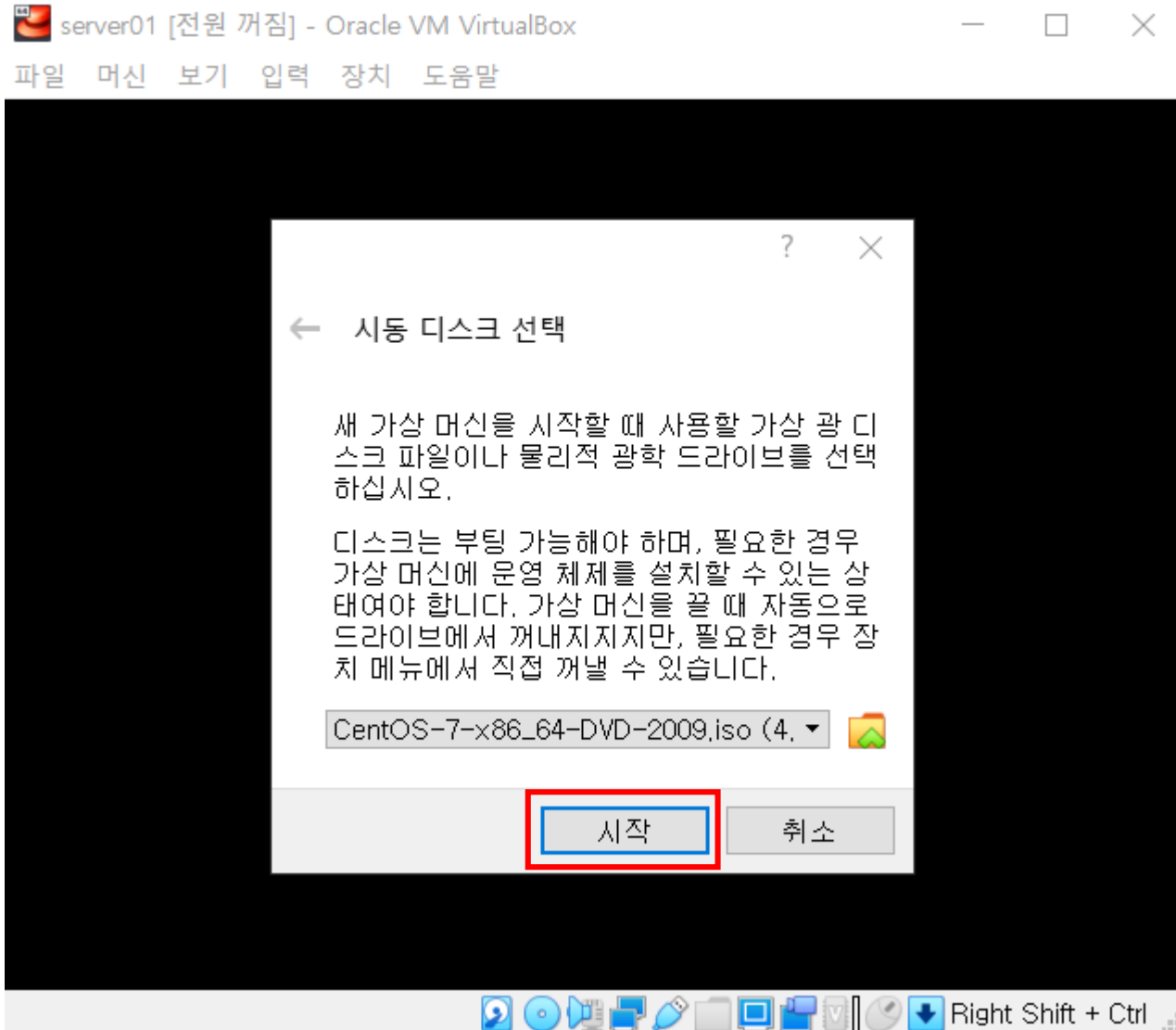
시스템
기본 메모리: 2048 MB
부팅 순서: 플로피, 광 디스크, 하드 디스크
가속: VT-X/AMD-V, 네스티드 페이징, PAE/NX, KVM 반가상화

미리 보기
server01

디스플레이
비디오 메모리: 16 MB
그래픽 컨트롤러: VMSVGA
원격 데스크톱 서버: 사용 안 함
녹화: 사용 안 함

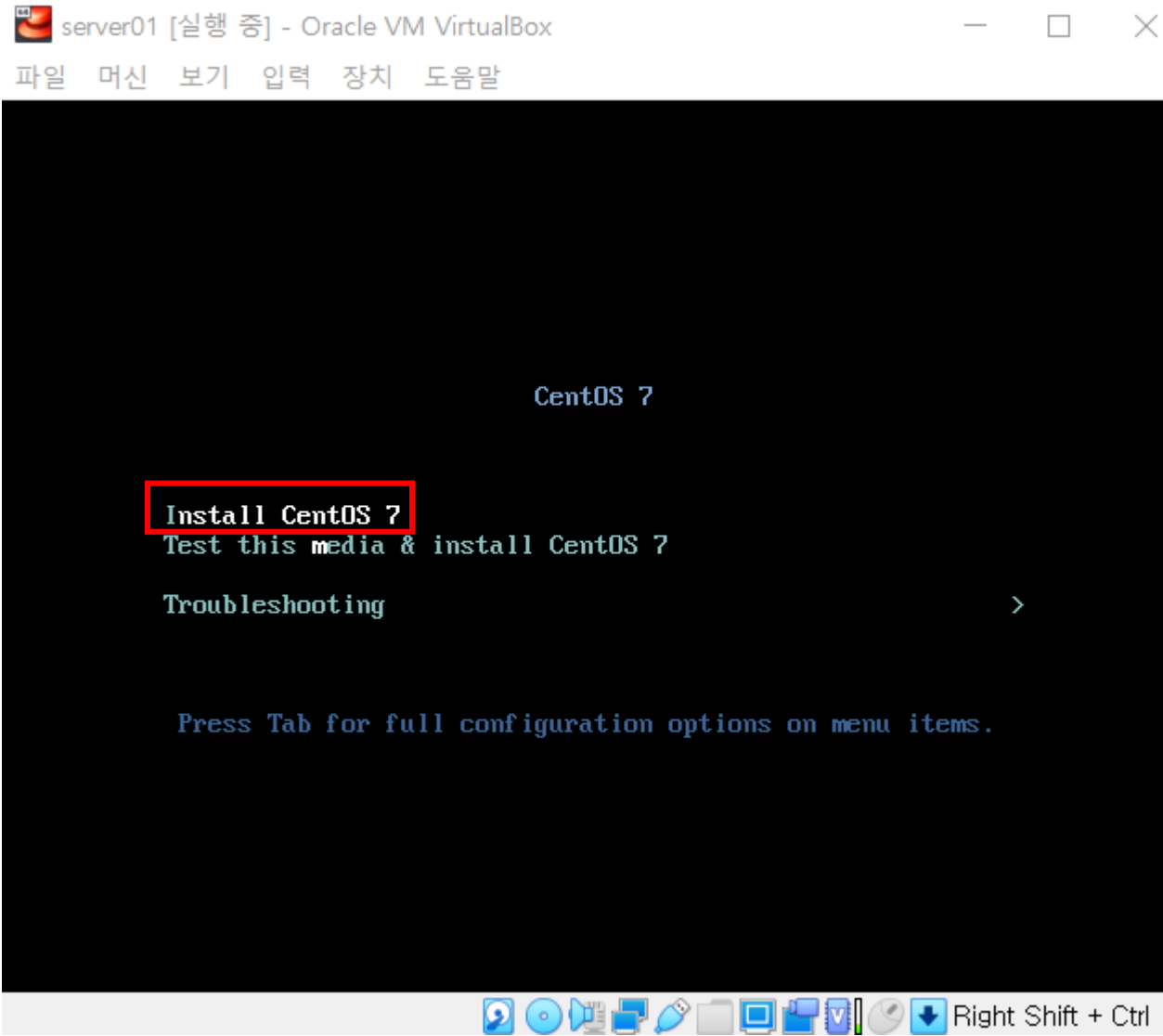
저장소
컨트롤러: IDE
IDE 세컨더리 마스터: [광학 드라이브] CentOS-7-x86_64-DVD-2009.iso (4.39 GB)
컨트롤러: SATA

2. 가상 서버 생성



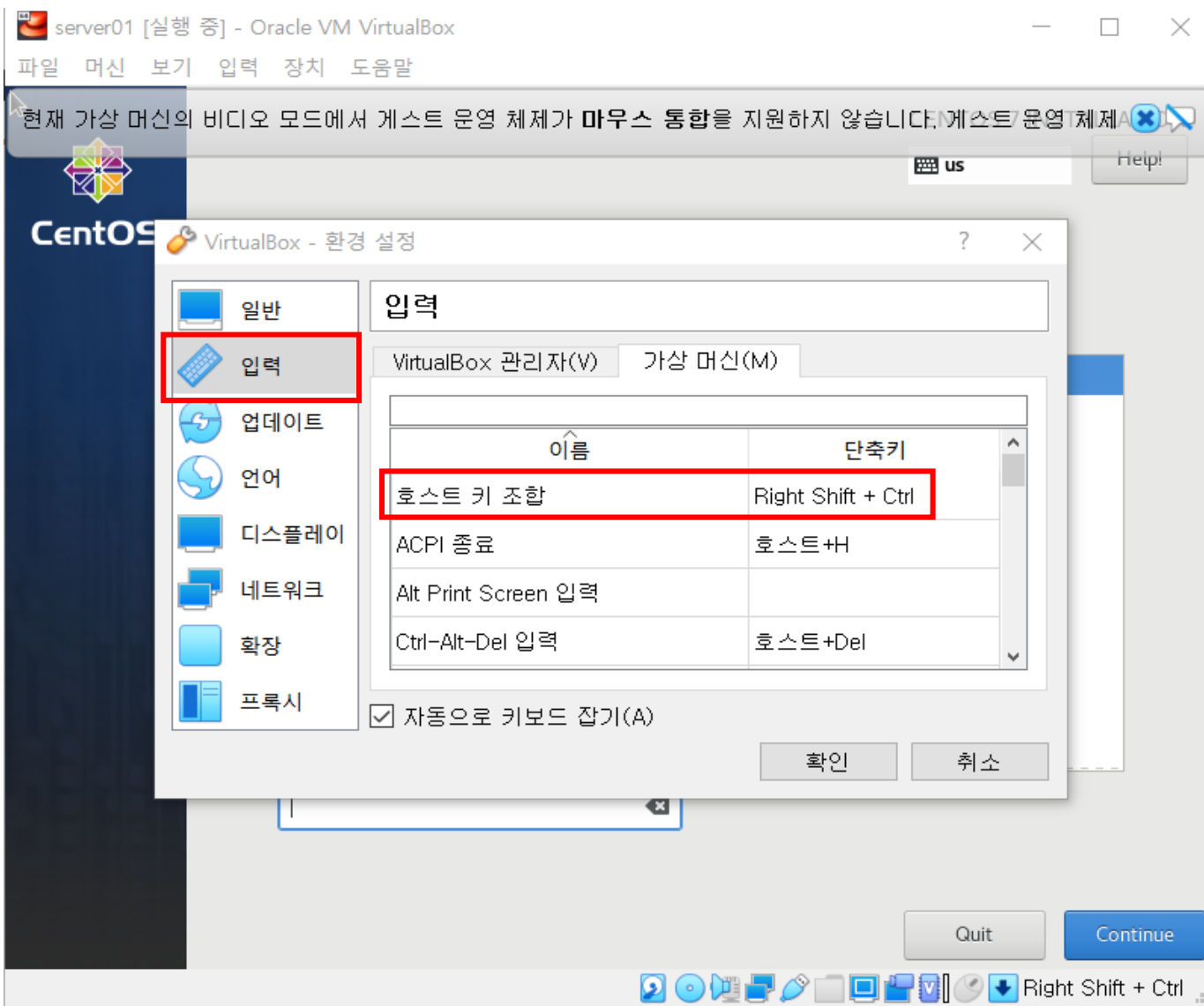
2. 가상 서버 생성

Install CentOS 7 Enter



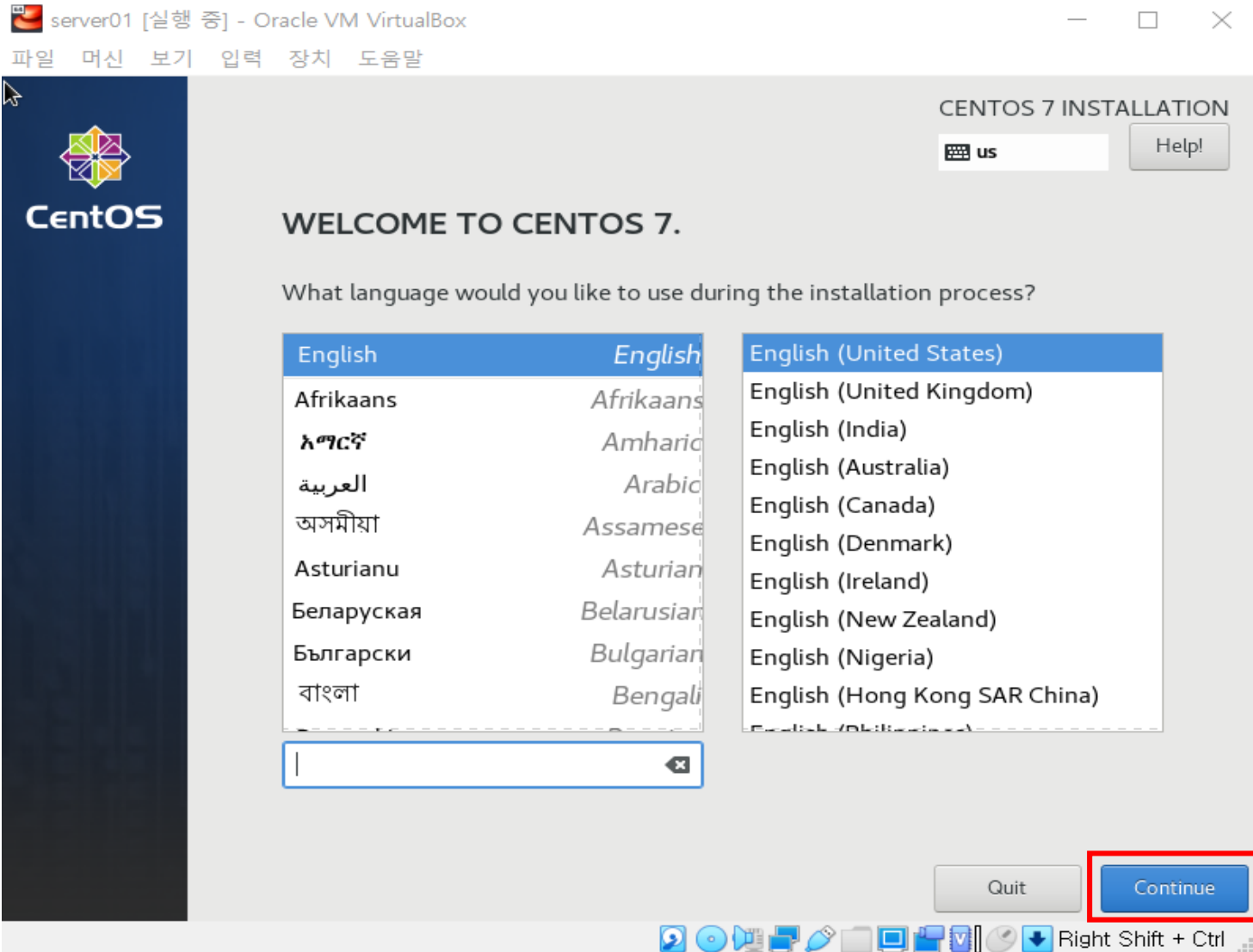
2. 가상 서버 생성

좌측 상단 파일 - 환경 설정 - 입력 - 호스트 키 조합 설정(게스트OS에서 호스트OS로 빠져나올 때 혹은 반대



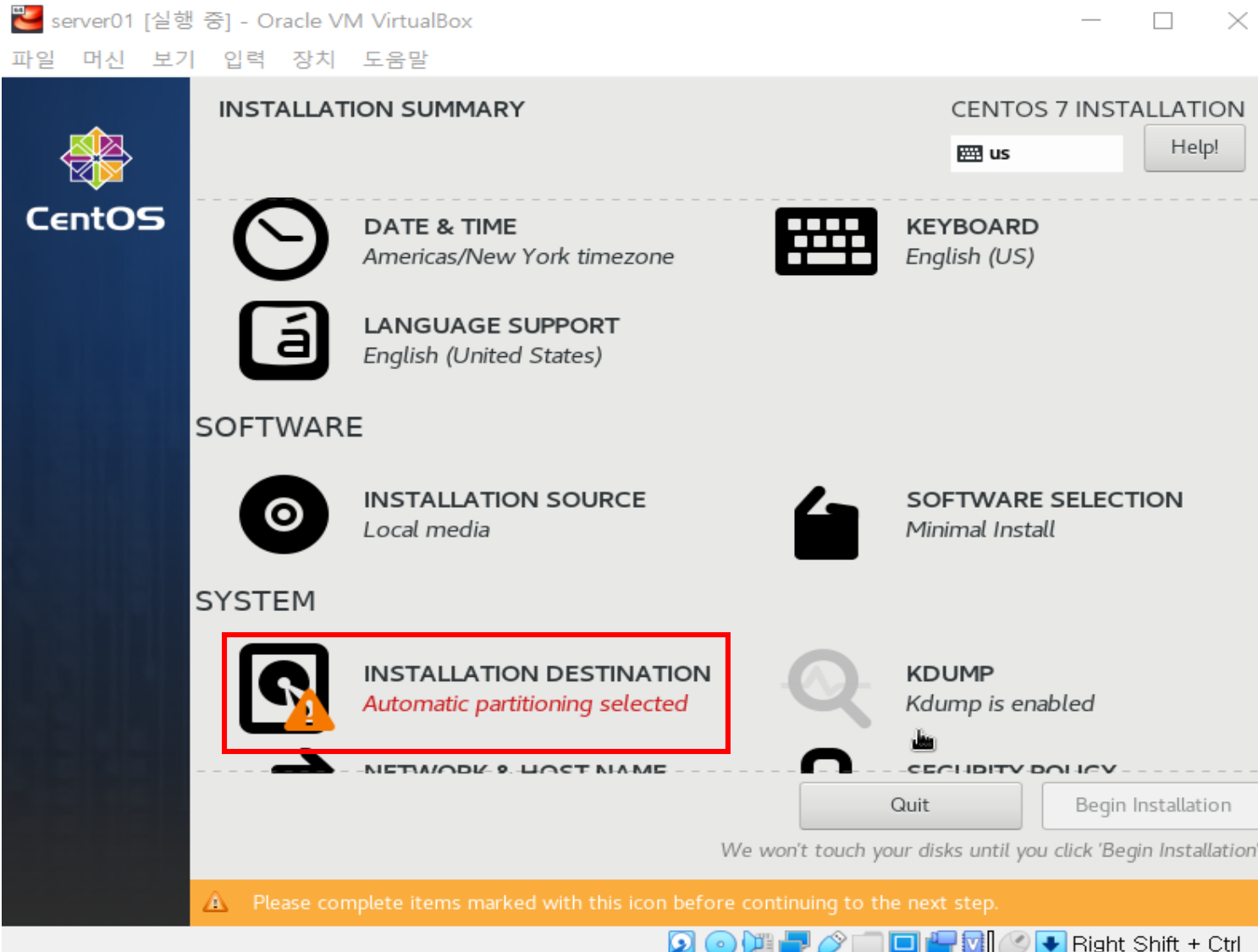
2. 가상 서버 생성

언어 설정 - Continue



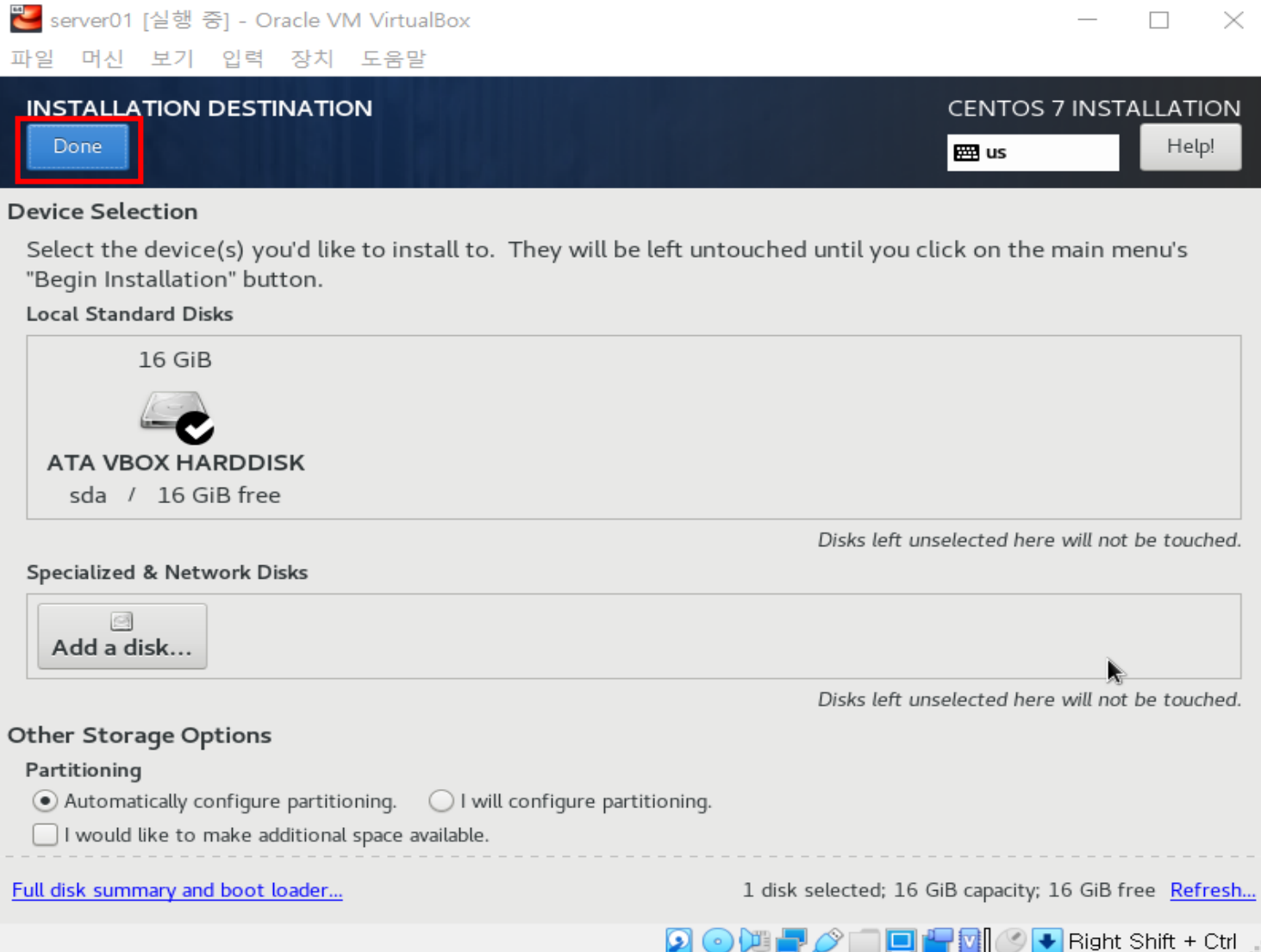
2. 가상 서버 생성

INSTALLATION DESTINATION 클릭



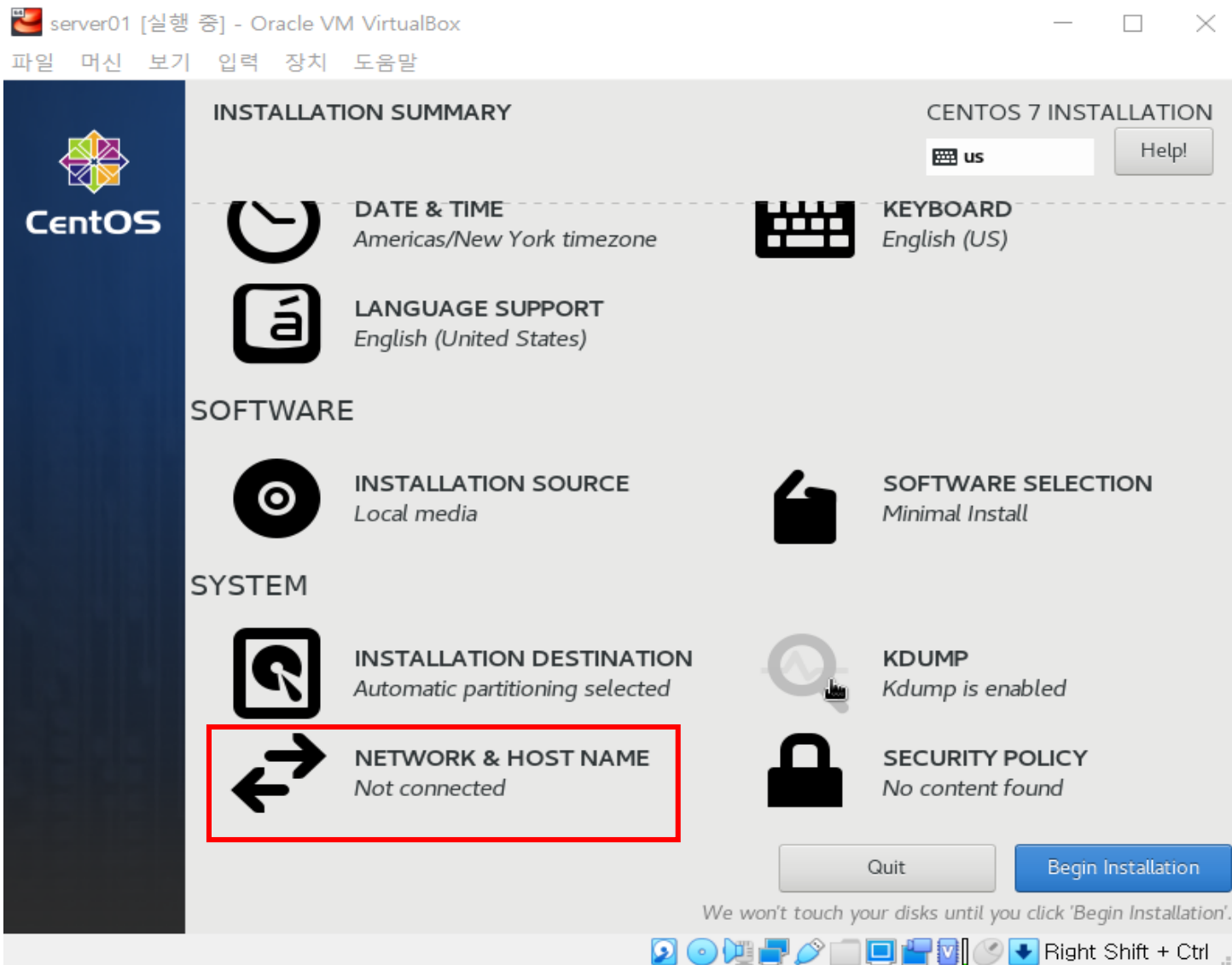
2. 가상 서버 생성

별다른 클릭 없이 Done 클릭 (* 경고 표시가 사라짐, 이 과정을 거치지 않아도 자동으로 설정됨)



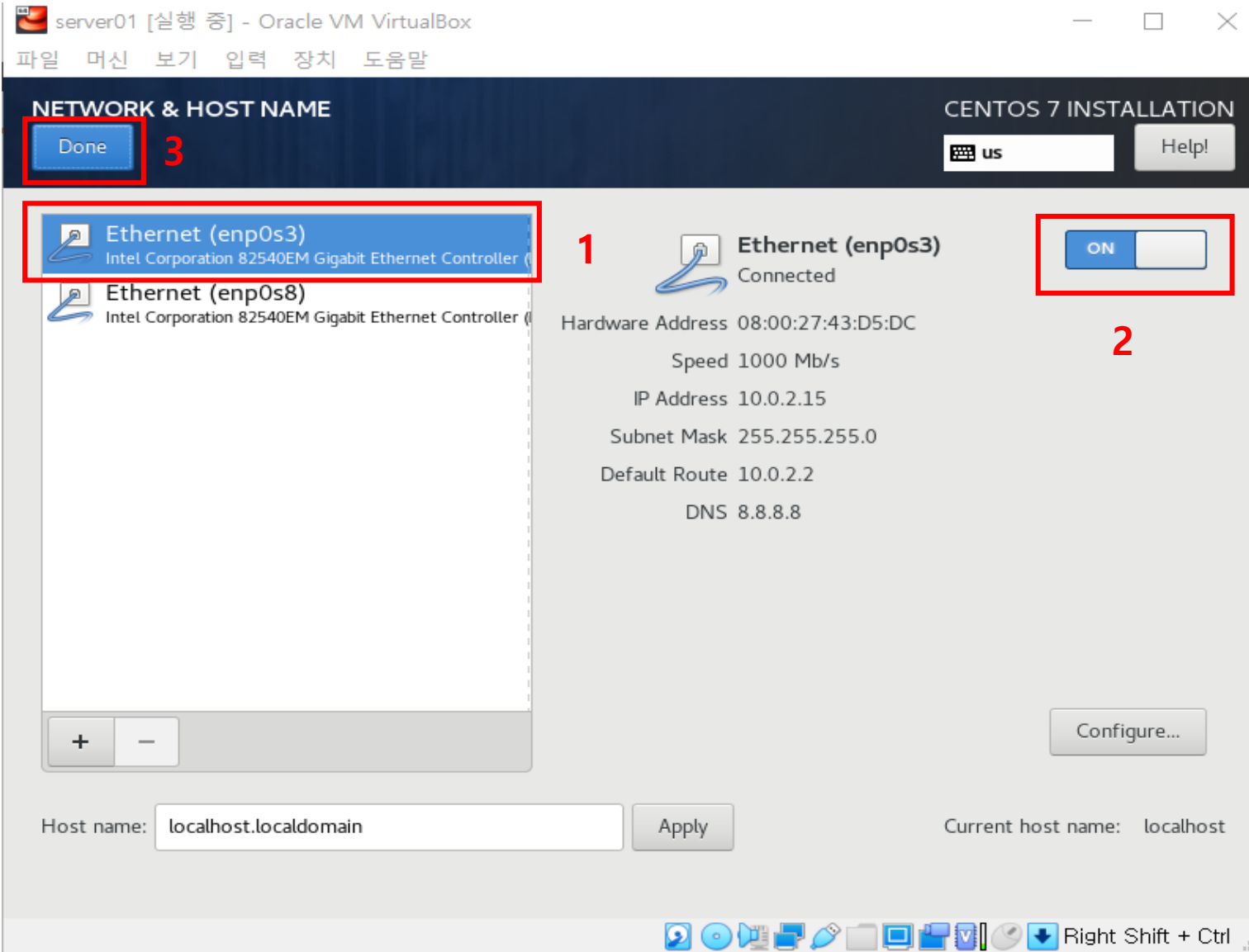
2. 가상 서버 생성

NETWORK & HOST NAME 클릭



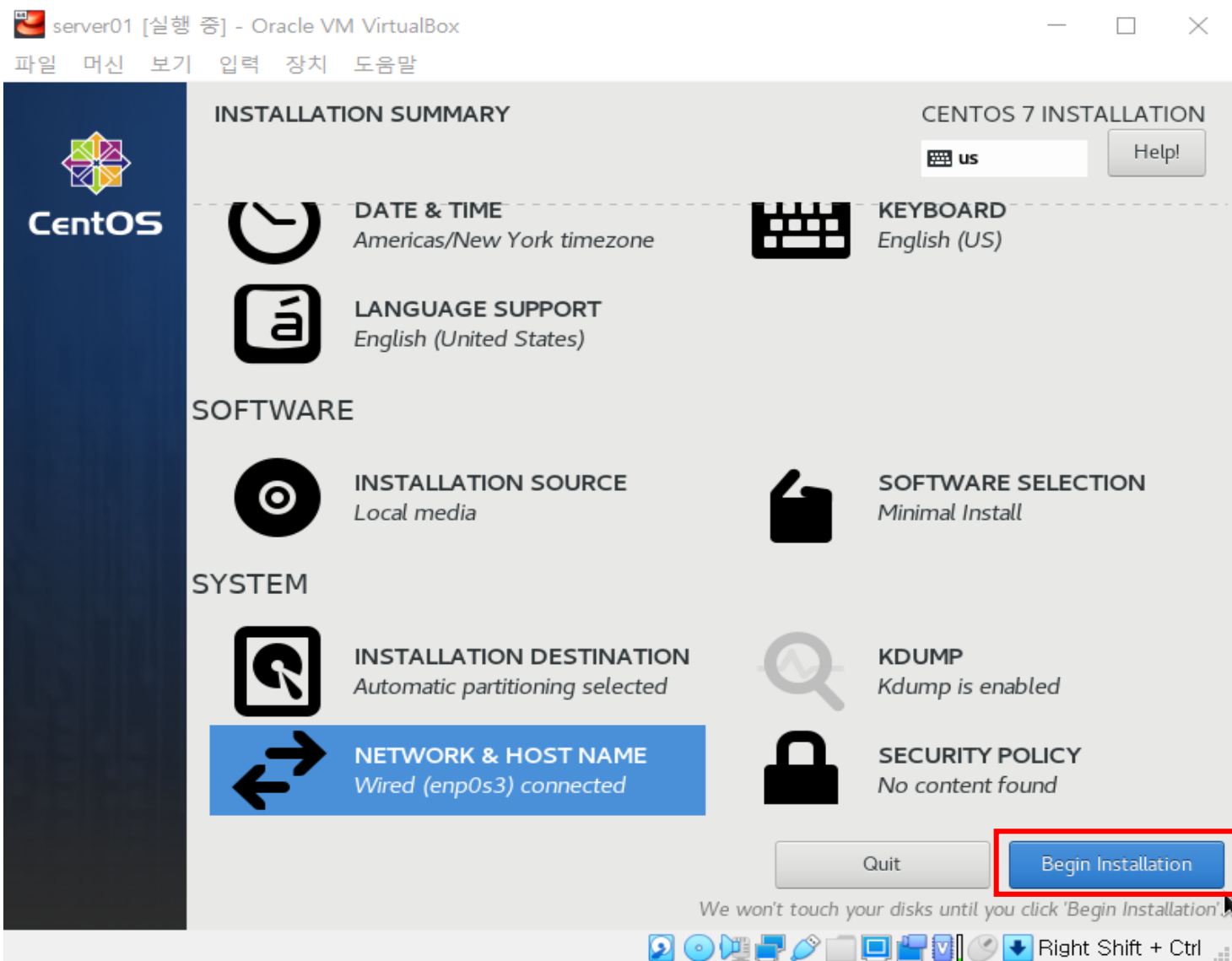
2. 가상 서버 생성

Enp0s3 이더넷 스위치 ON 클릭 후 Done (* enp0s8은 건드리지 않는다 – 서버 실행 후 설정할 예정)



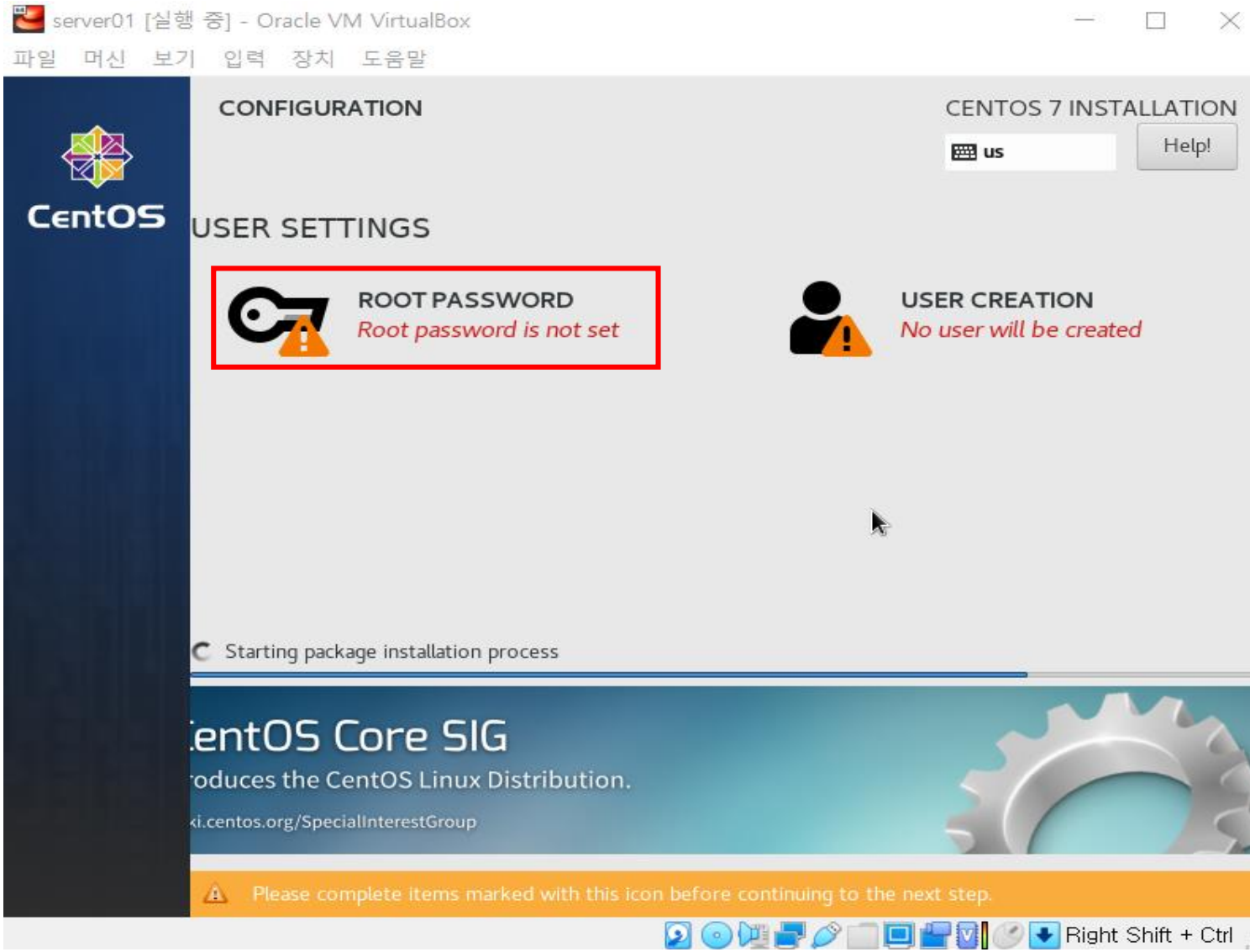
2. 가상 서버 생성

Begin Installation 클릭



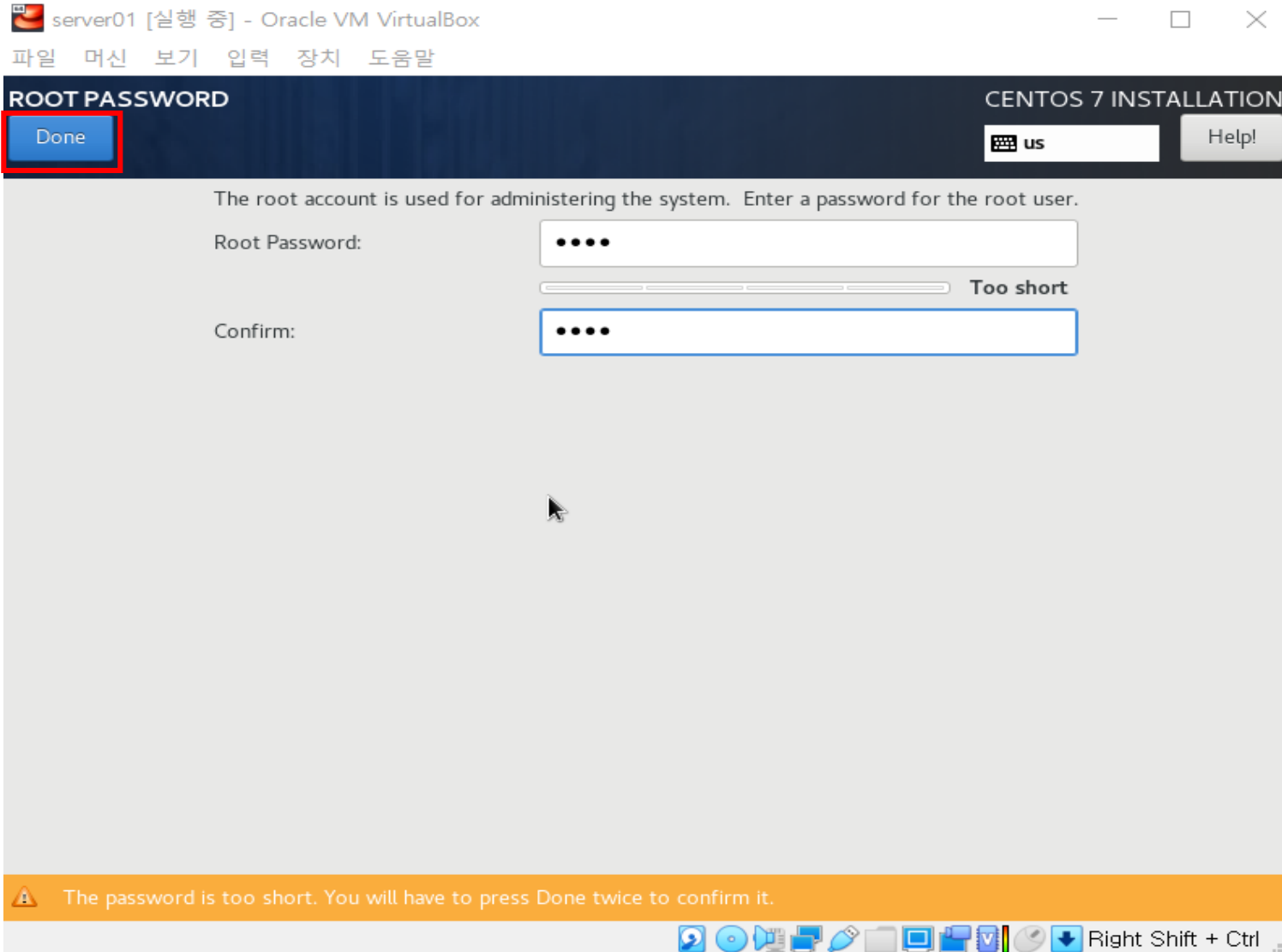
2. 가상 서버 생성

ROOT PASSWORD 클릭



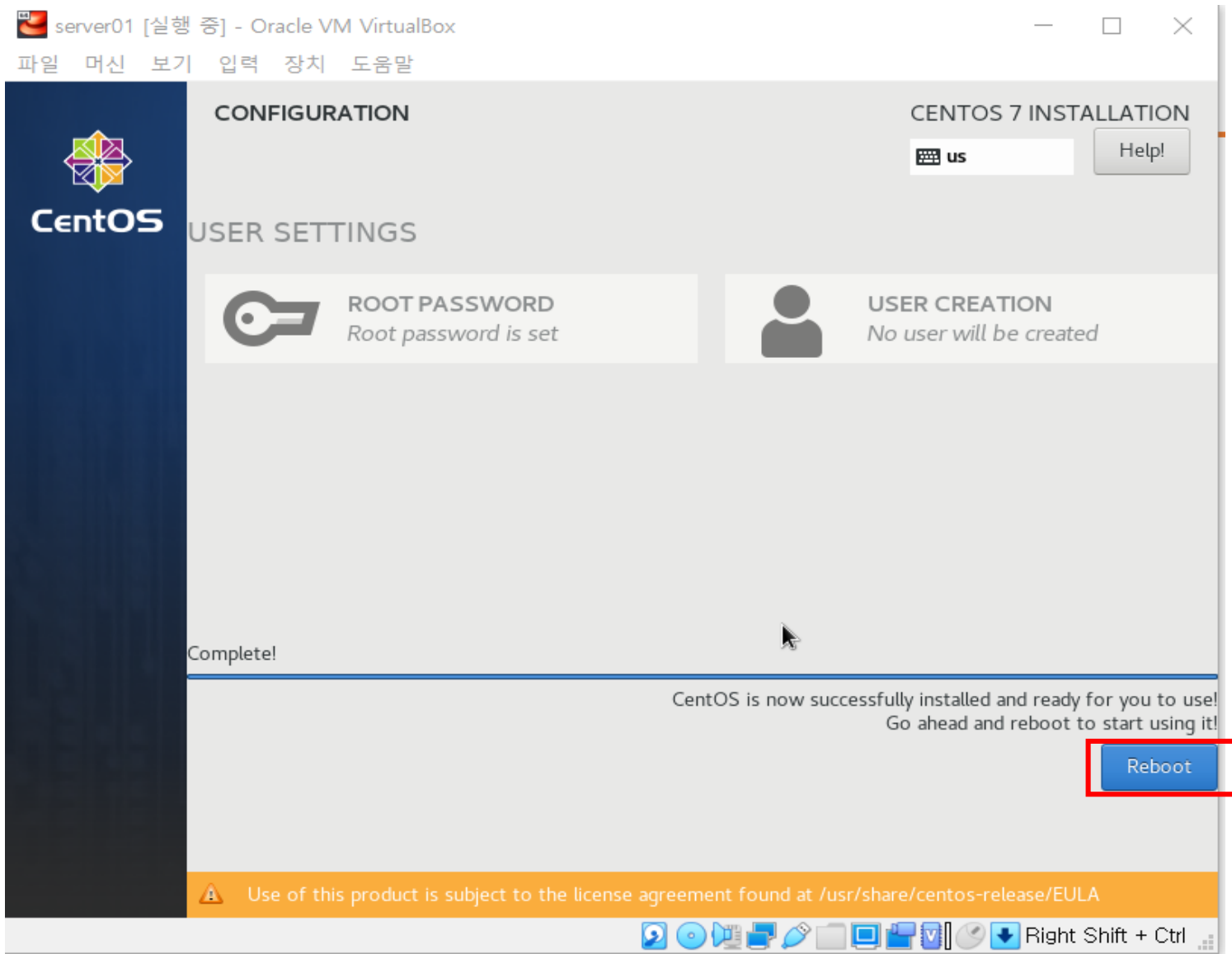
2. 가상 서버 생성

비밀번호 설정 후 Done (* 비밀번호가 권장하는 형식이 아니어도 Done을 두 번 클릭하면 설정되어 넘어간다).



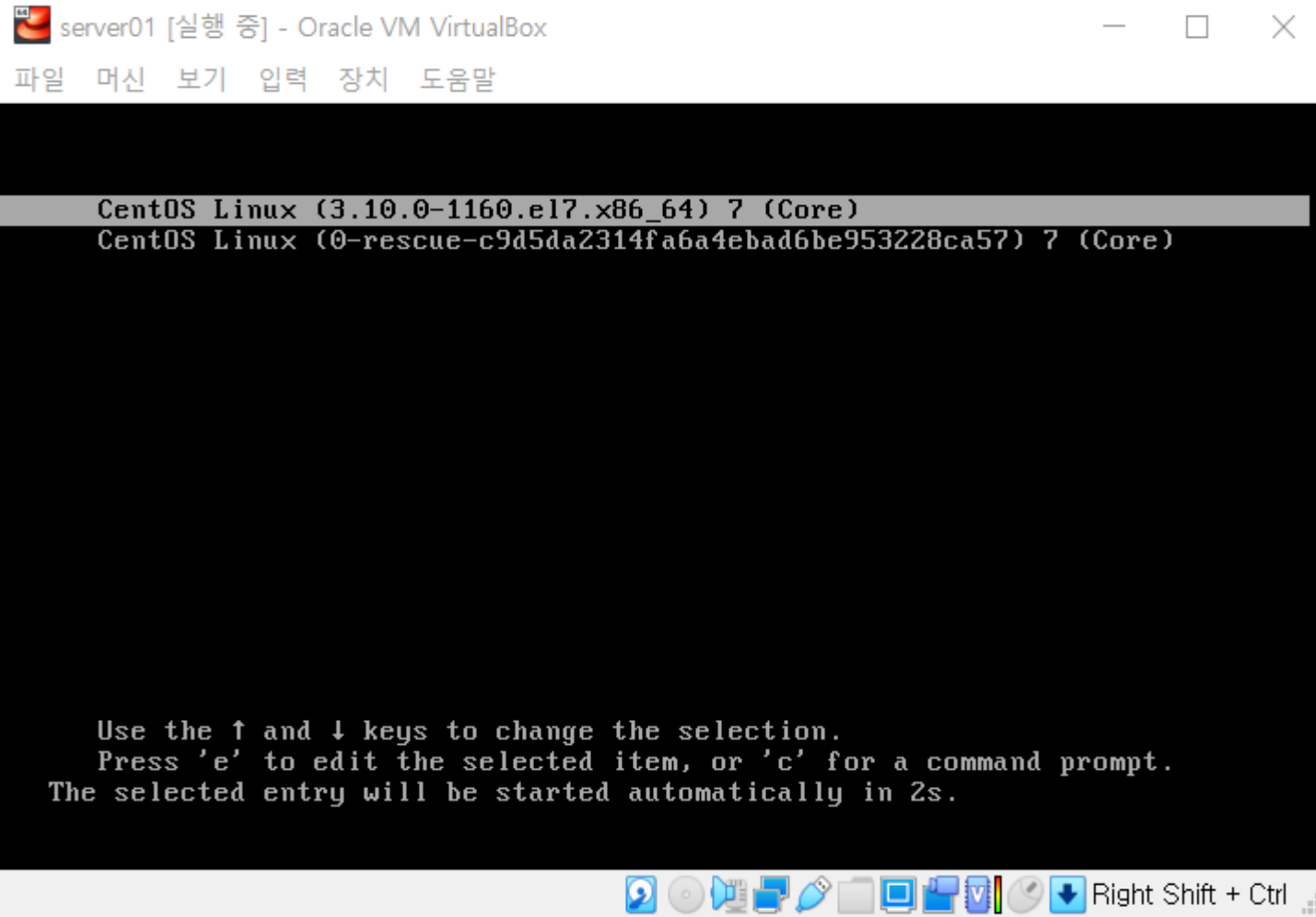
2. 가상 서버 생성

Reboot 클릭



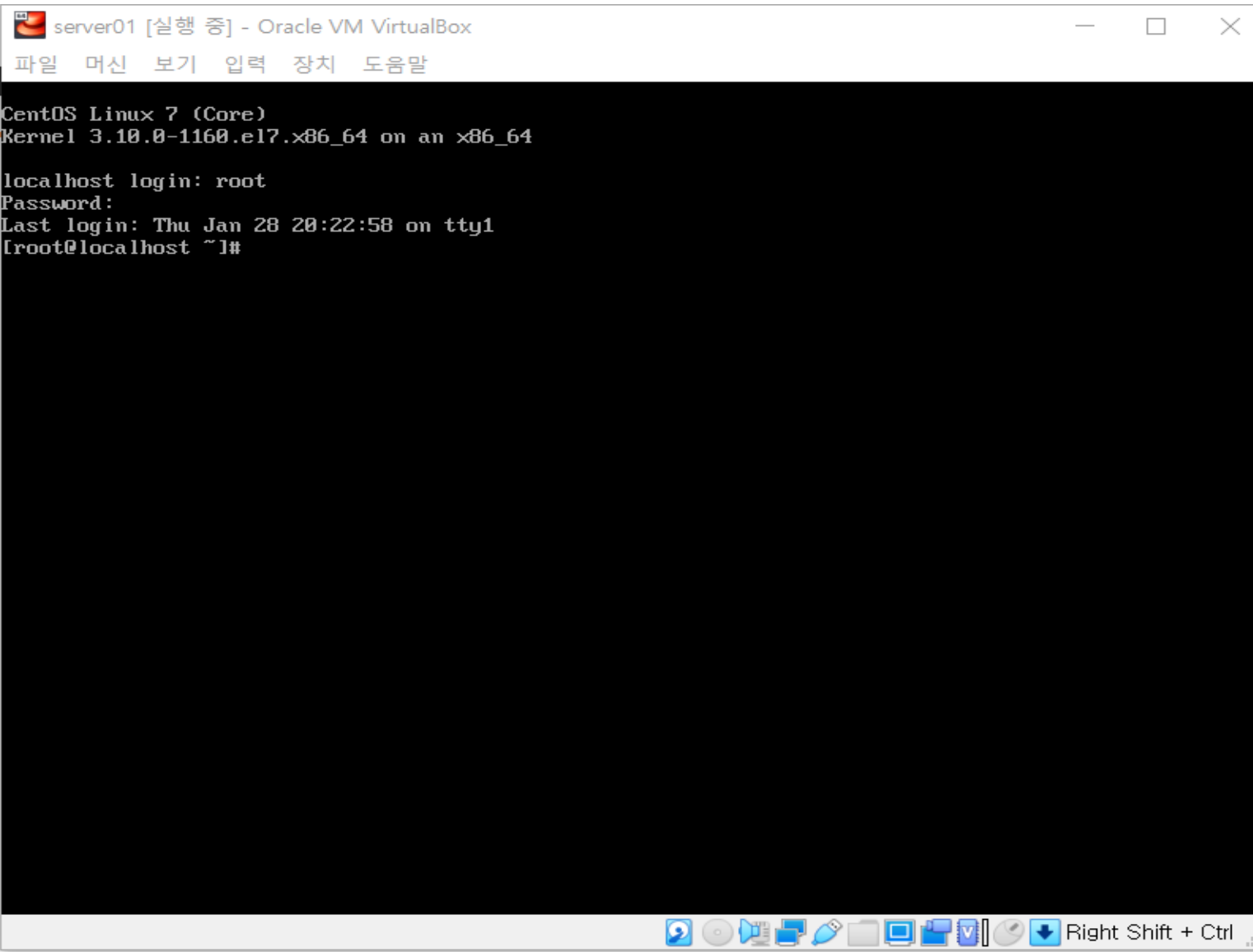
2. 가상 서버 생성

Enter



2. 가상 서버 생성

root – 비밀번호 입력 – 접속 확인



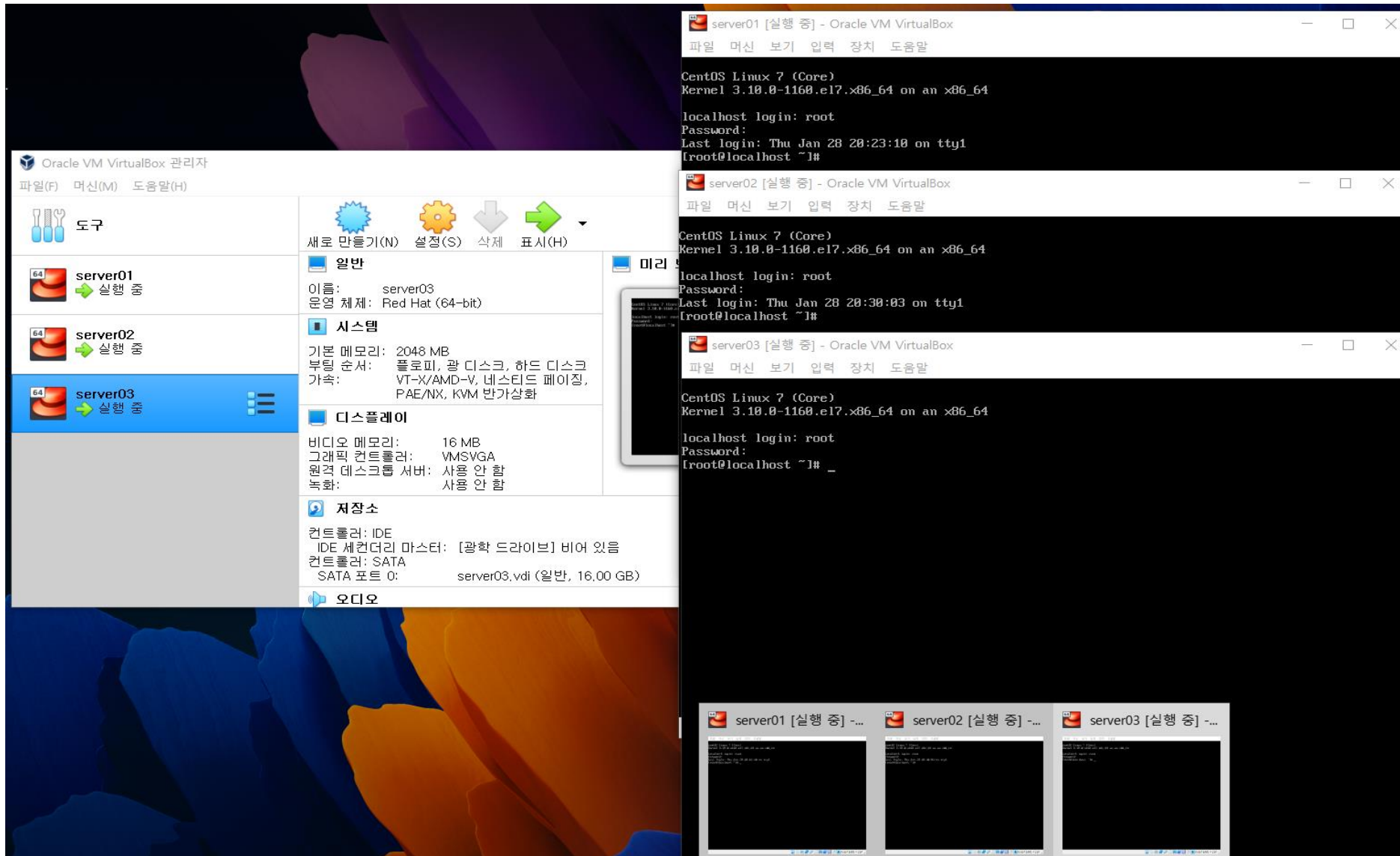
```
server01 [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

CentOS Linux 7 (Core)
Kernel 3.10.0-1160.el7.x86_64 on an x86_64

localhost login: root
Password:
Last login: Thu Jan 28 20:22:58 on tty1
[root@localhost ~]#
```

2. 가상 서버 생성

위 과정을 반복하여 3대의 서버를 구성한다. (server01, server02, server03)



3. 네트워크 설정

Action server: 명령을 수행할 서버

User: 명령을 수행할 계정

Pwd: 명령을 수행할 경로

3-1. 각 서버 별 hostname 변경

Action server: all

User: root

Pwd: /root

```
hostnamectl set-hostname [서버이름]
```

server01 [실행 중] - Oracle VM VirtualBox

파일 머신 보기 입력 장치 도움말

```
CentOS Linux 7 (Core)
Kernel 3.10.0-1160.el7.x86_64 on an x86_64

localhost login: root
Password:
Last login: Thu Jan 28 20:23:10 on tty1
[root@localhost ~]# hostnamectl set-hostname server01
[root@localhost ~]# _
```

server02 [실행 중] - Oracle VM VirtualBox

파일 머신 보기 입력 장치 도움말

```
CentOS Linux 7 (Core)
Kernel 3.10.0-1160.el7.x86_64 on an x86_64

localhost login: root
Password:
Last login: Thu Jan 28 20:30:03 on tty1
[root@localhost ~]# hostnamectl set-hostname server02
[root@localhost ~]#
```

server03 [실행 중] - Oracle VM VirtualBox

파일 머신 보기 입력 장치 도움말

```
[root@localhost ~]# hostnamectl set-hostname server03
[root@localhost ~]#
```

3-1. /etc/hosts 내용 변경

Action server: all

User: root

Pwd: /root

```
vi /etc/hosts
```

```
192.168.56.100 server01
```

```
192.168.56.101 server02
```

```
192.168.56.102 server03
```

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
```

```
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
```

```
192.168.56.100 server01
```

```
192.168.56.101 server02
```

```
192.168.56.102 server03
```

3-2. 고정 IP 할당 (enp0s8)

Action server: all

User: root

Pwd: /root

```
cat /etc/sysconfig/network-scripts/ifcfg-enp0s3
```

ONBOOT="yes"만 확인(다른 설정 수정 X)

```
vi /etc/sysconfig/network-scripts/ifcfg-enp0s8
```

수정 후

```
systemctl restart network
```

```
ip addr show
```

enp-0s3

```
TYPE="Ethernet"  
PROXY_METHOD="none"  
BROWSER_ONLY="no"  
BOOTPROTO="dhcp"  
DEFROUTE="yes"  
IPV4_FAILURE_FATAL="no"  
IPV6INIT="yes"  
IPV6_AUTOCONF="yes"  
IPV6_DEFROUTE="yes"  
IPV6_FAILURE_FATAL="no"  
IPV6_ADDR_GEN_MODE="stable-privacy"  
NAME="enp0s3"  
UUID="db98795f-36b0-47f9-b670-898355b70f3f"  
DEVICE="enp0s3"  
ONBOOT="yes"  
[root@server01 ~]#
```

ONBOOT만 확인

다른 설정을 그대로 놔두기

enp-0s8

```
PROXY_METHOD=none  
BROWSER_ONLY=no  
BOOTPROTO=none  
DEFROUTE=yes  
IPV4_FAILURE_FATAL=no  
IPV6INIT=yes  
IPV6_AUTOCONF=yes  
IPV6_DEFROUTE=yes  
IPV6_FAILURE_FATAL=no  
IPV6_ADDR_GEN_MODE=stable-privacy  
NAME=enp0s8  
UUID=b5ce7b4f-de03-4e42-9e44-707e4a3671b6  
DEVICE=enp0s8  
ONBOOT=yes  
IPADDR=192.168.56.100  
GATEWAY=192.168.56.1  
DNS1=192.168.56.1  
PREFIX=24
```

```
systemctl restart network
```

BOOTPROTO=none: dhcp 비활성화

ONBOOT=yes: 스위치 on

IPADDR: 서버의 고정 IP

(* server01: 192.168.56.100
server02: 192.168.56.101
server03: 192.168.56.102)

GATEWAY: 호스트OS가 할당한 Virtual box IP

DNS1: 동일

PREFIX: 넷마스크와 비슷한 의미

* IPADDR만 각 서버별로 다름

* enp0s3는 초기 설정 때 스위치ON 클릭 시 설정
할 필요가 없고, 스위치ON을 하지 않았다면
ONBOOT만 yes로 변경

확인 ip addr show

server01

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:43:d5:dc brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global noprefixroute dynamic enp0s3
        valid_lft 86189sec preferred_lft 86189sec
    inet6 fe80::11dc:6ba5:2df2:e675/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:73:1d:18 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.100/24 brd 192.168.56.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::1770:f4fd:f8a8:b03f/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

server02

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:34:b3:73 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global noprefixroute dynamic enp0s3
        valid_lft 86168sec preferred_lft 86168sec
    inet6 fe80::28e0:3e1d:8f21:a3fc/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:26:58:e1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::783:abdd:a4b7:fd0e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```


server03

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:2a:6e:24 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global noprefixroute dynamic enp0s3
        valid_lft 86162sec preferred_lft 86162sec
    inet6 fe80::428d:facc:5ff6:576d/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:18:d1:54 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::b13b:949b:c4ed:6f03/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

3-3. 재부팅 후 핑 테스트

Action server: all

User: root

Pwd: /root

reboot

ping -c 2 google.com

모든 서버 hostname 변경과 외부 핑 테스트

```
[root@server01 ~]# hostname  
server01
```

```
[root@server01 ~]# ping -c 2 google.com
```

```
PING google.com (216.58.196.238) 56(84) bytes of data.
```

```
64 bytes from kix06s01-in-f238.1e100.net (216.58.196.238): icmp_seq=1 ttl=113 time=58.6 ms
```

```
64 bytes from kix06s01-in-f238.1e100.net (216.58.196.238): icmp_seq=2 ttl=113 time=56.3 ms
```

```
--- google.com ping statistics ---
```

```
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
```

```
rtt min/avg/max/mdev = 56.314/57.488/58.663/1.198 ms
```

```
[root@server01 ~]# _
```

4.SELINUX 해제

Action server: all

User: root

Pwd: /root

```
vi /etc/sysconfig/selinux
```

```
SELINUX=disabled
```

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

SELINUXTYPE를 잘못 설정하게 되면
서버 부팅이 안됨 주의!

5. 방화벽해제

Action server: all

User: root

Pwd: /root

```
systemctl stop firewalld.service
```

```
systemctl disable firewalld.service
```

```
systemctl restart network
```

```
firewall-cmd --state
```

```
not running
```

모든 서버 방화벽 해제 확인

```
[root@server01 ~]# systemctl stop firewalld.service
[root@server01 ~]# systemctl disable firewalld.service
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@server01 ~]# systemctl restart network
[root@server01 ~]# firewall-cmd --state
not running
[root@server01 ~]# _
```

6.모듈 설치

Action server: all

User: root

Pwd: /root

```
yum update
```

```
yum install wget
```

```
yum install ntp
```

7.ntp 설정

Action server: all

User: root

Pwd: /root

```
vi /etc/ntp.conf
```

[모든 서버]

```
#server 0.centos.pool.ntp.org iburst  
#server 1.centos.pool.ntp.org iburst  
#server 2.centos.pool.ntp.org iburst  
#server 3.centos.pool.ntp.org iburst  
주석처리
```

[server01]

```
restrict 192.168.1.0 mask 255.255.0.0 nomodify notrap (주석 해제)
```

```
server 127.127.1.0
```


[server02]

restrict 192.168.1.0 mask 255.255.0.0 nomodify notrap (주석 해제)

server server01

server 127.127.1.0

[server03]

restrict 192.168.1.0 mask 255.255.0.0 nomodify notrap (주석 해제)

server server01

server server02

[server01]

```
# Hosts on local network are less restricted.  
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap  
  
# Use public servers from the pool.ntp.org project.  
# Please consider joining the pool (http://www.pool.ntp.org/join.html).  
#server 0.centos.pool.ntp.org iburst  
#server 1.centos.pool.ntp.org iburst  
#server 2.centos.pool.ntp.org iburst  
#server 3.centos.pool.ntp.org iburst  
server 127.127.1.0  
  
#broadcast 192.168.1.255 autokey      # broadcast server  
#broadcastclient                      # broadcast client  
#broadcast 224.0.1.1 autokey          # multicast server  
#multicastclient 224.0.1.1            # multicast client  
#mancastserver 239.255.254.254        # manycast server  
#mancastclient 239.255.254.254 autokey # manycast client
```

[server02]

```
# Hosts on local network are less restricted.  
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap  
  
# Use public servers from the pool.ntp.org project.  
# Please consider joining the pool (http://www.pool.ntp.org/join.html).  
#server 0.centos.pool.ntp.org iburst  
#server 1.centos.pool.ntp.org iburst  
#server 2.centos.pool.ntp.org iburst  
#server 3.centos.pool.ntp.org iburst  
  
server 127.127.1.0  
server server01  
  
#broadcast 192.168.1.255 autokey      # broadcast server  
#broadcastclient                      # broadcast client  
#broadcast 224.0.1.1 autokey          # multicast server  
#multicastclient 224.0.1.1            # multicast client  
#mcastserver 239.255.254.254          # mcast server  
#mcastclient 239.255.254.254 autokey  # mcast client
```

[server03]

```
# Hosts on local network are less restricted.  
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap  
  
# Use public servers from the pool.ntp.org project.  
# Please consider joining the pool (http://www.pool.ntp.org/join.html).  
#server 0.centos.pool.ntp.org iburst  
#server 1.centos.pool.ntp.org iburst  
#server 2.centos.pool.ntp.org iburst  
#server 3.centos.pool.ntp.org iburst  
  
server server01  
server server02  
  
#broadcast 192.168.1.255 autokey      # broadcast server  
#broadcastclient                      # broadcast client  
#broadcast 224.0.1.1 autokey          # multicast server  
#multicastclient 224.0.1.1            # multicast client  
#manycastserver 239.255.254.254       # manycast server  
#manycastclient 239.255.254.254 autokey # manycast client
```

7.ntp 등록, 시작, 확인

Action server: all

User: root

Pwd: /root

```
systemctl enable ntpd
```

```
systemctl start ntpd
```

```
ntpq -p
```

[server01]

```
[root@server01 ~]# ntpq -p
      remote           refid      st t when poll reach  delay  offset  jitter
=====
*LOCAL(0)        .LOCL.          5 l  35  64   1   0.000   0.000   0.000
[root@server01 ~]#
```

[server02]

```
[root@server02 ~]# ntpq -p
      remote           refid      st t when poll reach  delay  offset  jitter
=====
*LOCAL(0)        .LOCL.          5 l   6  64   3   0.000   0.000   0.000
server01         LOCAL(0)        6 u   4  64   3   0.737  -0.635   0.183
[root@server02 ~]#
```

[server03]

```
[root@server03 ~]# ntpq -p
      remote           refid      st t when poll reach  delay  offset  jitter
=====
server01         LOCAL(0)        6 u  12  64   3   0.709  -0.179   0.165
server02         LOCAL(0)        6 u   9  64   3   0.668   0.541   0.249
[root@server03 ~]# _
```

8.ssh 통신

8-1.ssh 키 생성

Action server: all

User: root

Pwd: /root

ssh-keygen

* 계속해서 Enter

```
Generating public/private rsa key pair.  
Enter file in which to save the key (/root/.ssh/id_rsa):  
Created directory '/root/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /root/.ssh/id_rsa.  
Your public key has been saved in /root/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:1HidkAuTX/W81sEjtbRSBmLn1RUTtLMAsfXPGnn+4WM root@server02  
The key's randomart image is:  
+---[RSA 2048]---+  
|                 ... *o+oQ*|  
|      +..+o.Boo *|  
|    .++oo. =oX  |  
|      oo      +=*|  
|       S       =.=|  
|               . =|  
|               ...|  
|               .Eo|  
|               ..o|  
+---[SHA256]---+  
[root@server02 ~]#
```

8-2.ssh 키 공유

Action server: all

User: root

Pwd: /root

```
ssh-copy-id -i .ssh/id_rsa.pub root@server01
```

```
ssh-copy-id -i .ssh/id_rsa.pub root@server02
```

```
ssh-copy-id -i .ssh/id_rsa.pub root@server03
```

* 툴이 없을 시 `yum install -y openssh-clients`

* 모든 서버에서 위 명령들 실행해야 함

```
ssh server01
```

```
exit
```

```
ssh server02
```

```
exit
```

```
ssh server03
```

```
exit
```

비밀번호 여부 없이 접속되는지 확인

server02, server03에서도 다른 서버들이 접속되는지 확인

```
[root@server01 ~]# ssh server01
Last login: Thu Jan 28 23:15:15 2021 from gateway
[root@server01 ~]# exit
logout
Connection to server01 closed.
[root@server01 ~]# ssh server02
Last login: Thu Jan 28 23:15:31 2021 from gateway
[root@server02 ~]# exit
logout
Connection to server02 closed.
[root@server01 ~]# ssh server03
Last login: Thu Jan 28 23:15:39 2021 from gateway
[root@server03 ~]# exit
logout
Connection to server03 closed.
[root@server01 ~]#
```

9.MobaXterm 시작

MobaXterm: putty, xshell 등과 같은 서버 접속 툴

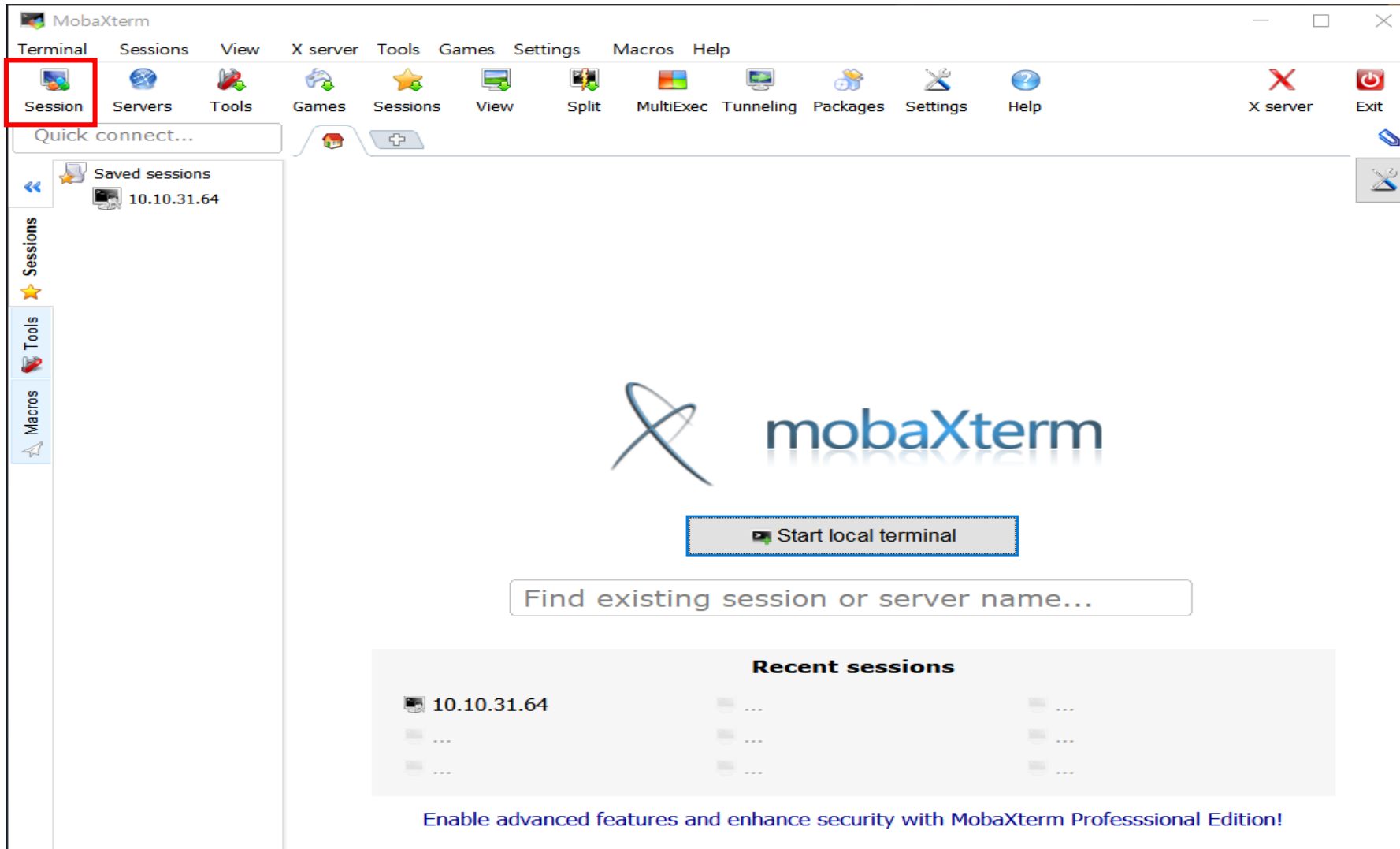
여러 서버를 한 눈에 보고 작업, sftp를 통해 로컬의 파일들을 편리하게 배포 가능

다운로드 받았던 MobaXterm 시작

(* virtual box의 실행 중인 서버는 끄면 안된다. 끄게 될 시 MobaXterm과도 연결이 끊김

9-1.MobaXterm과 가상서버 ssh 연결

MobaXterm 실행 – 좌측 상단 Session



서버 IP 입력 후 OK (* 3대 서버 모두 연결)

Session settings



Basic SSH settings

Remote host * 192.168.56.100

☐ Specify username



Port 22



Advanced SSH settings

Terminal settings

Network settings

Bookmark settings

Secure Shell (SSH) session

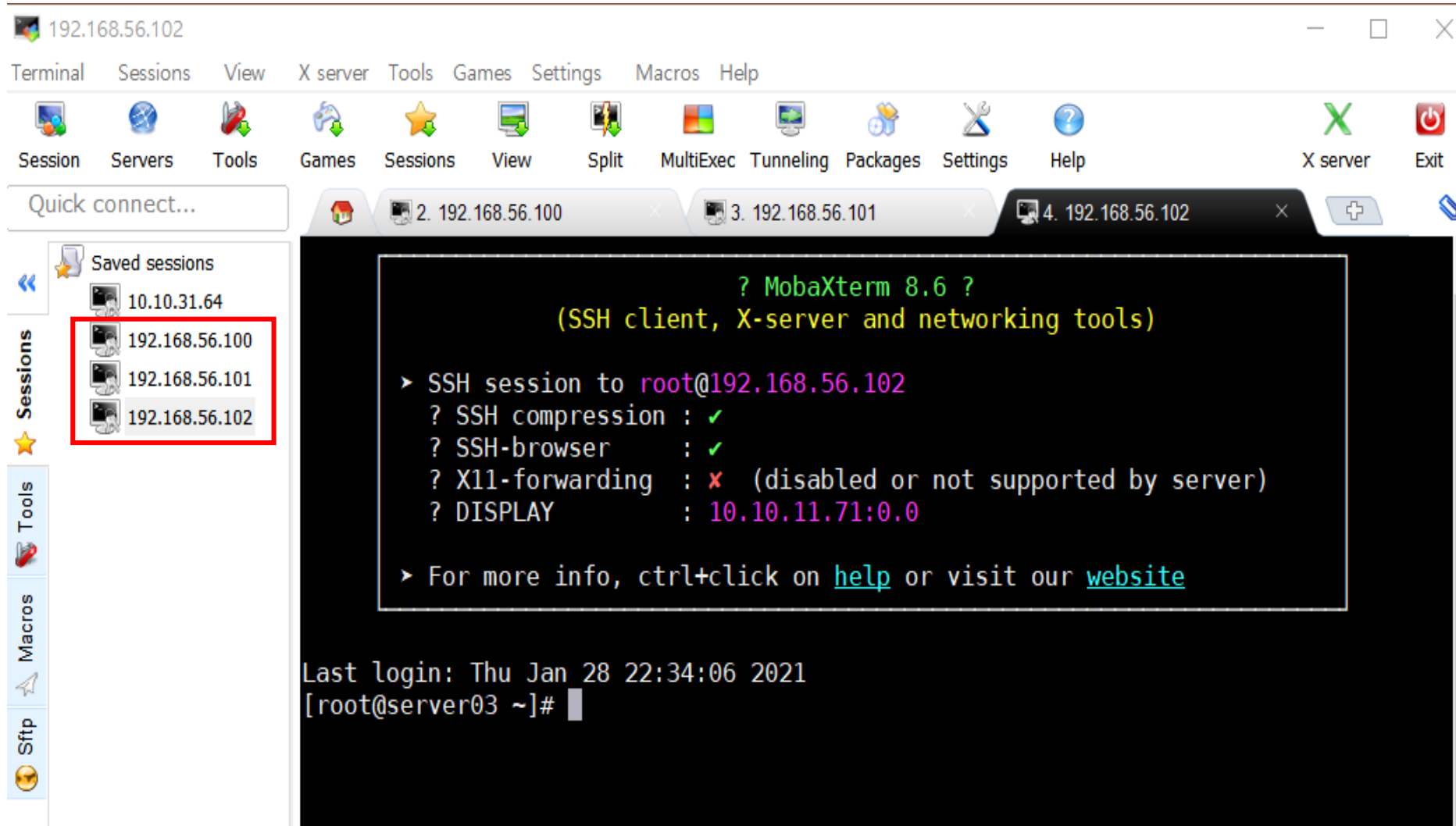


OK

Cancel

결과

(* 이후 과정들은 MobaXterm으로 작업, virtual box의 서버들은 실행 상태로 유지



10. JDK 배포 및 설정

MobaXterm의 Sftp 기능을 사용해 다운로드 받은 jdk 압축 폴더를 손쉽게 배포

10-1. jdk 설치 여부 확인

Action server: all

User: root

Pwd: /root

```
java -version #자바 설치 여부 확인
```

```
rpm -qa | grep jdk #설치되어 있다면 설치된 jdk 정보 확인
```

```
yum remove [현재 jdk]
```

```
java -version #삭제되었는지 확인
```

10-2. JDK 설치 폴더 생성

Action server: all

User: root

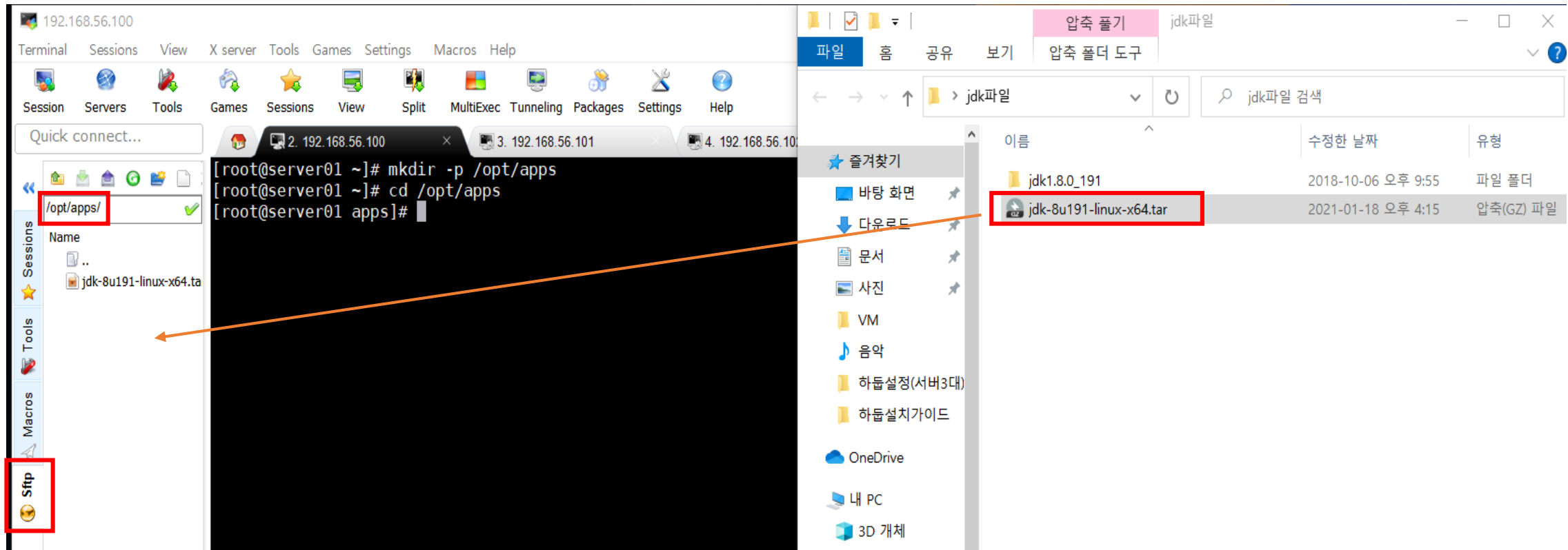
Pwd: /root

```
mkdir -p /opt/apps #폴더 생성
```

```
cd /opt/apps
```

10-3. JDK 압축 파일 옮기기

좌측 Sftp 클릭 – 경로 /opt/apps 입력 후 Enter – 다운로드 받은 jdk 압축파일 드래그
(* 모든 서버 동일)



10-4. JDK 압축 해제 및 폴더명 변경

Action server: all
User: root
Pwd: /opt/apps

```
ls
```

```
tar xvfz jdk-8u191-linux-x64.tar.gz #이름에 맞게
```

```
ls
```

```
mv Jdk1.8.0_191 jdk8 #폴더명 변경, 가독성 위함
```

```
[root@server01 apps]# ls
jdk-8u191-linux-x64.tar.gz
[root@server01 apps]# tar xvfz jdk-8u191-linux-x64.tar.gz
```

```
[root@server01 apps]# ls
jdk1.8.0_191  jdk-8u191-linux-x64.tar.gz
[root@server01 apps]# mv jdk1.8.0_191 jdk8
[root@server01 apps]# ls
jdk8  jdk-8u191-linux-x64.tar.gz
[root@server01 apps]#
```

10-5. JDK 환경 변수 설정

Action server: all

User: root

Pwd: /root

```
cd ~ #root로 경로 변경
```

```
vi /etc/profile.d/java.sh
```

```
export JAVA_HOME=/opt/apps/jdk8
```

```
export PATH=$PATH:$JAVA_HOME/bin #추가 (* ctrl+c 하고 vi 에디터 내에서 shift+insert 하면 붙여넣기 가능)
```

```
source /etc/profile.d/java.sh #변경 내역 적용
```

```
java -version #자바 실행 확인
```

모든 서버 확인

```
export JAVA_HOME=/opt/apps/jdk8  
export PATH=$PATH:$JAVA_HOME/bin
```

```
[root@server01 ~]# vi /etc/profile.d/java.sh  
[root@server01 ~]# source /etc/profile.d/java.sh  
[root@server01 ~]# java -version  
java version "1.8.0_191"  
Java(TM) SE Runtime Environment (build 1.8.0_191-b12)  
Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode)  
[root@server01 ~]#
```

11. 계정 생성 (zookeeper, Hadoop)

Action server: all

User: root

Pwd: /root

```
adduser zookeeper #계정 추가
```

```
passwd zookeeper #비밀번호 설정 (편의상 비밀번호도 zookeeper)
```

```
adduser hadoop #계정 추가
```

```
passwd hadoop #비밀번호 설정 (편의상 비밀번호도 hadoop)
```

11-1. HDFS 여러 정보 저장 폴더 생성 및 하둡 계정에 권한 부여

Action server: all

User: root

Pwd: /root

```
mkdir /dfs  
mkdir /yarn  
mkdir /pids
```

```
chown hadoop /dfs  
chgrp hadoop /dfs  
chown hadoop /yarn  
chgrp hadoop /yarn  
chown hadoop /pids  
chgrp hadoop /pids
```

```
mkdir /dfs  
mkdir /yarn  
mkdir /pids  
chown hadoop /dfs  
chgrp hadoop /dfs  
chown hadoop /yarn  
chgrp hadoop /yarn  
chown hadoop /pids  
chgrp hadoop /pids
```

```
ls -l
```

```
bin  dev  etc  lib  media  opt  proc  run  srv  tmp  var  
boot dfs  home lib64 mnt  pids  root  sbin sys  usr  yarn
```

12.zookeeper ssh 통신

12-1.zookeeper 계정 접속

Action server: all
User: root
Pwd: /root

```
su - zookeeper
```

12-2.zookeeper ssh키 생성

Action server: all

User: zookeeper

Pwd: /zookeeper/home

```
ssh-keygen
```

12-3.zookeeper ssh키 공유

Action server: all

User: zookeeper

Pwd: /zookeeper/home


```
ssh-copy-id -i .ssh/id_rsa.pub zookeeper@server01
```

```
ssh-copy-id -i .ssh/id_rsa.pub zookeeper@server01
```

```
ssh-copy-id -i .ssh/id_rsa.pub zookeeper@server01
```

```
ssh server01
```

```
ssh server02
```

```
ssh server03 #각 서버 별 접속 확인
```

```
*ssh 배포 과정은 위 root계정 ssh 배포과정과 동일
```

13.zookeeper 설치

Action server: server01 (* server01에서 선 설치, 파일 수정 후 재압축하여 server02, server03에게 배포)

User: zookeeper

Pwd: /home/zookeeper

```
wget https://archive.apache.org/dist/zookeeper/zookeeper-3.4.10/zookeeper-3.4.10.tar.gz
```

13-1.zookeeper 압축 파일 해제

Action server: server01

User: zookeeper

Pwd: /home/zookeeper

```
tar xvfz zookeeper-3.4.10.tar.gz
```

```
ls
```

```
[zookeeper@server01 ~]$ ls
zookeeper-3.4.10  zookeeper-3.4.10.tar.gz
[zookeeper@server01 ~]$
```

13-2.zookeeper config 파일 수정

Action server: server01

User: zookeeper

Pwd: /home/zookeeper

```
cd zookeeper-3.4.10 #경로 이동
```

```
cp conf/zoo_sample.cfg conf/zoo.cfg #파일 복사
```

```
vi conf/zoo.cfg #파일 수정
```

```
tickTime=2000
```

```
initLimit=10
```

```
syncLimit=5
```

```
dataDir=/home/zookeeper/data
```

```
clientPort=2181
```

```
maxClientCnxns=0
```

```
maxSessionTimeout=180000
```

```
server.1=server01:2888:3888
```

```
server.2=server02:2888:3888
```

```
server.3=server03:2888:3888
```

```
tickTime=2000  
initLimit=10  
syncLimit=5  
dataDir=/home/zookeeper/data  
clientPort=2181  
maxClientCnxns=0  
maxSessionTimeout=180000  
server.1=server01:2888:3888  
server.2=server02:2888:3888  
server.3=server03:2888:3888
```

13-2.zookeeper 폴더 재압축

Action server: server01

User: zookeeper

Pwd: /home/zookeeper

```
cd .. #경로 이동 -> pwd home/zookeeper
```

```
ls
```

```
tar cvfz zookeeper.tar.gz zookeeper-3.4.10 #파일 설정 후 재압축
```

```
ls
```

```
[zookeeper@server01 ~]$ ls  
zookeeper-3.4.10  zookeeper-3.4.10.tar.gz  zookeeper.tar.gz  
[zookeeper@server01 ~]$
```

13-3.zookeeper 재압축 파일 배포

Action server: server01

User: zookeeper

Pwd: /home/zookeeper

```
scp zookeeper.tar.gz zookeeper@server02:/home/zookeeper #배포
```

```
scp zookeeper.tar.gz zookeeper@server03:/home/zookeeper #오류 시 ssh 설정 다시 확인
```

13-4.zookeeper 재압축 파일 해제

Action server: server02, server03

User: zookeeper

Pwd: /home/zookeeper

```
tar xvfz zookeeper.tar.gz #server02, server03에서 압축 해제
```

13-5.zookeeper myid 지정

Action server: all

User: zookeeper

Pwd: /home/zookeeper

```
mkdir data #data 폴더 생성
```

```
cd data #경로 이동
```

```
vi myid #id 생성 server01은 1, server02은 2, server03은 3
```

```
cd ~ #돌아오기
```



```
[zookeeper@server01 ~]$ cat data/myid  
1  
[zookeeper@server01 ~]$
```

```
[zookeeper@server02 ~]$ cat data/myid  
2  
[zookeeper@server02 ~]$
```

```
[zookeeper@server03 ~]$ cat data/myid  
3  
[zookeeper@server03 ~]$
```

13-6.zookeeper 서버 실행

Action server: all

User: zookeeper

Pwd: /home/zookeeper

```
cd zookeeper-3.4.10 #경로 이동
```

```
./bin/zkServer.sh start #주키퍼 서버 실행
```

13-7.zookeeper 서버 상태 확인

Action server: all

User: zookeeper

Pwd: /home/zookeeper/zookeeper-3.4.10

```
./bin/zkServer.sh status #서버 별 상태 확인
```

```
exit #root 계정으로 변경
```

```
[zookeeper@server01 zookeeper-3.4.10]$ ./bin/zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /home/zookeeper/zookeeper-3.4.10/bin/../conf/zoo.cfg
Mode: follower
[zookeeper@server01 zookeeper-3.4.10]$
```

```
[zookeeper@server02 zookeeper-3.4.10]$ ./bin/zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /home/zookeeper/zookeeper-3.4.10/bin/../conf/zoo.cfg
Mode: follower
[zookeeper@server02 zookeeper-3.4.10]$
```

```
[zookeeper@server03 zookeeper-3.4.10]$ ./bin/zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /home/zookeeper/zookeeper-3.4.10/bin/../conf/zoo.cfg
Mode: leader
[zookeeper@server03 zookeeper-3.4.10]$
```

한 서버는 leader, 두 서버는 follower

14.hadoop ssh 통신

14-1.hadoop 계정 접속(13-7이후 root계정으로 접속해 있어야 함)

Action server: all

User: root

Pwd: /root

```
su - hadoop
```

14-2.ssh 키 생성

Action server: all
User: hadoop
Pwd: /home/hadoop

```
ssh-keygen #ssh 키 생성
```

14-3.ssh 키 공유

Action server: all
User: hadoop
Pwd: /home/hadoop

```
ssh-copy-id -i .ssh/id_rsa.pub hadoop@server01  
  
ssh-copy-id -i .ssh/id_rsa.pub hadoop@server02  
  
ssh-copy-id -i .ssh/id_rsa.pub hadoop@server03
```

```
ssh server01 #접속 확인  
exit
```

```
ssh server02  
exit
```

```
ssh server03  
exit
```

*ssh키 배포 과정은 위 root, zookeeper ssh 배포 과정과 동일

15.hadoop 설치

Action server: server01 (*주키퍼와 마찬가지로 한 서버에서 파일 수정 후 재압축 및 배포)
User: hadoop
Pwd: /home/hadoop

```
wget https://archive.apache.org/dist/hadoop/core/hadoop-3.1.0/hadoop-3.1.0.tar.gz
```


15-1.hadoop 압축파일 해제

Action server: server01

User: hadoop

Pwd: /home/hadoop

```
tar xvfz hadoop-3.1.0.tar.gz
```

```
ls
```

```
[hadoop@server01 ~]$ ls
hadoop-3.1.0  hadoop-3.1.0.tar.gz
[hadoop@server01 ~]$
```

15-2.hadoop 설정 파일 경로 이동

Action server: server01

User: hadoop

Pwd: /home/hadoop

```
cd hadoop-3.1.0/etc/hadoop
```

15-3.hadoop 파일 설정

(workers, core-site.xml, hdfs-site.xml, mapred-site.xml, yarn-site.xml, hadoop-env.sh)

Action server: server01

User: hadoop

Pwd: /home/hadoop/hadoop-3.1.0/etc/hadoop

workers

* 노드로 사용할 호스트명 입력 (하둡2버전에서는 slave 파일명, datanode, nodemanager

```
vi workers
```

```
server01  
server02  
server03
```

```
#localhost  
server01  
server02  
server03
```

Action server: server01

User: hadoop

Pwd: /home/hadoop/hadoop-3.1.0/etc/hadoop

core-site.xml

* HDFS와 맵리듀스에서 공통으로 사용할 환경 설정

vi core-site.xml

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://NNHA</value>
  </property>
  <property>
    <name>ha.zookeeper.quorum</name>
    <value>server01:2181,server02:2181,server03:2181</value>
  </property>
</configuration>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
```

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://NNHA</value>
  </property>
  <property>
    <name>ha.zookeeper.quorum</name>
    <value>server01:2181,server02:2181,server03:2181</value>
  </property>
</configuration>
```

Action server: server01

User: hadoop

Pwd: /home/hadoop/hadoop-3.1.0/etc/hadoop

hdfs-site.xml

* HDFS에서 사용할 환경 설정

vi hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.client.failover.proxy.provider.NNHA</name>
    <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>/dfs/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/dfs/datanode</value>
  </property>
  <property>
    <name>dfs.journalnode.edits.dir</name>
    <value>/dfs/journalnode</value>
  </property>
  <property>
    <name>dfs.nameservices</name>
    <value>NNHA</value>
  </property>
```



```
<property>
  <name>dfs.ha.namenodes.NNHA</name>
  <value>server01,server02</value>
</property>
<property>
  <name>dfs.namenode.rpc-address.NNHA.server01</name>
  <value>server01:8020</value>
</property>
<property>
  <name>dfs.namenode.rpc-address.NNHA.server02</name>
  <value>server02:8020</value>
</property>
<property>
  <name>dfs.namenode.http-address.NNHA.server01</name>
  <value>server01:9870</value>
</property>
<property>
  <name>dfs.namenode.http-address.NNHA.server02</name>
  <value>server02:9870</value>
</property>
<property>
  <name>dfs.namenode.shared.edits.dir</name>
  <value>qjournal://server01:8485;server02:8485;server03:8485/NNHA</value>
</property>
```

```
<property>
  <name>dfs.ha.automatic-failover.enabled</name>
  <value>true</value>
</property>
<property>
  <name>dfs.ha.fencing.methods</name>
  <value>sshfence</value>
</property>
<property>
  <name>dfs.ha.fencing.ssh.private-key-files</name>
  <value>/home/hadoop/.ssh/id_rsa</value>
</property>
</configuration>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->
```

```
<configuration>
  <property>
    <name>dfs.client.failover.proxy.provider.NNHA</name>
    <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>/dfs/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/dfs/datanode</value>
  </property>
  <property>
    <name>dfs.journalnode.edits.dir</name>
    <value>/dfs/journalnode</value>
  </property>
  <property>
    <name>dfs.nameservices</name>
    <value>NNHA</value>
  </property>
  <property>
    <name>dfs.ha.namenodes.NNHA</name>
    <value>server01,server02</value>
```

```
</property>
<property>
  <name>dfs.namenode.rpc-address.NNHA.server01</name>
  <value>server01:8020</value>
</property>
<property>
  <name>dfs.namenode.rpc-address.NNHA.server02</name>
  <value>server02:8020</value>
</property>
<property>
  <name>dfs.namenode.http-address.NNHA.server01</name>
  <value>server01:9870</value>
</property>
<property>
  <name>dfs.namenode.http-address.NNHA.server02</name>
  <value>server02:9870</value>
</property>
<property>
  <name>dfs.namenode.shared.edits.dir</name>
  <value>qjournal://server01:8485;server02:8485;server03:8485/NNHA</value>
</property>
<property>
  <name>dfs.ha.automatic-failover.enabled</name>
  <value>true</value>
</property>
<property>
  <name>dfs.ha.fencing.methods</name>
  <value>sshfence</value>
</property>
<property>
  <name>dfs.ha.fencing.ssh.private-key-files</name>
  <value>/home/hadoop/.ssh/id_rsa</value>
</property>
</configuration>
```

Action server: server01

User: hadoop

Pwd: /home/hadoop/hadoop-3.1.0/etc/hadoop

mapred-site.xml

* 맵리듀스에서 사용할 환경 설정

vi mapred-site.xml

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->
```

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

Action server: server01

User: hadoop

Pwd: /home/hadoop/hadoop-3.1.0/etc/hadoop

yarn-site.xml

* yarn에서 사용할 환경 설정

vi 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.nodemanager.local-dirs</name>
    <value>/yarn/nm-local-dir</value>
  </property>
  <property>
    <name>yarn.resourcemanager.fs.state-store.uri</name>
    <value>/yarn/system/rmstore</value>
  </property>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>server01</value>
  </property>
```



```
<property>
  <name>yarn.resourcemanager.address</name>
  <value>server01:8032</value>
</property>
<property>
  <name>yarn.web-proxy.address</name>
  <value>0.0.0.0:8089</value>
</property>
<property>
  <name>yarn.application.classpath</name>
  <value>
    /home/hadoop/hadoop-3.1.0/etc/hadoop,
    /home/hadoop/hadoop-3.1.0/share/hadoop/common/*,
    /home/hadoop/hadoop-3.1.0/share/hadoop/common/lib/*,
    /home/hadoop/hadoop-3.1.0/share/hadoop/hdfs/*,
    /home/hadoop/hadoop-3.1.0/share/hadoop/hdfs/lib/*,
    /home/hadoop/hadoop-3.1.0/share/hadoop/mapreduce/*,
    /home/hadoop/hadoop-3.1.0/share/hadoop/mapreduce/lib/*,
    /home/hadoop/hadoop-3.1.0/share/hadoop/yarn/*,
    /home/hadoop/hadoop-3.1.0/share/hadoop/yarn/lib/*
  </value>
</property>
</configuration>
```

```
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
```

```
<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.nodemanager.local-dirs</name>
    <value>/yarn/nm-local-dir</value>
  </property>
  <property>
    <name>yarn.resourcemanager.fs.state-store.uri</name>
    <value>/yarn/system/rmstore</value>
  </property>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>server01</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>server01:8032</value>
  </property>
  <property>
    <name>yarn.web-proxy.address</name>
```

```
        <value>0.0.0.0:8089</value>
    </property>
    <property>
        <name>yarn.application.classpath</name>
        <value>
            /home/hadoop/hadoop-3.1.0/etc/hadoop,
            /home/hadoop/hadoop-3.1.0/share/hadoop/common/*,
            /home/hadoop/hadoop-3.1.0/share/hadoop/common/lib/*,
            /home/hadoop/hadoop-3.1.0/share/hadoop/hdfs/*,
            /home/hadoop/hadoop-3.1.0/share/hadoop/hdfs/lib/*,
            /home/hadoop/hadoop-3.1.0/share/hadoop/mapreduce/*,
            /home/hadoop/hadoop-3.1.0/share/hadoop/mapreduce/lib/*,
            /home/hadoop/hadoop-3.1.0/share/hadoop/yarn/*,
            /home/hadoop/hadoop-3.1.0/share/hadoop/yarn/lib/*
        </value>
    </property>
</configuration>
```

Action server: server01

User: hadoop

Pwd: /home/hadoop/hadoop-3.1.0/etc/hadoop

hadoop-env.sh

* jdk와 pids경로 설정

vi hadoop-env.sh

[54번째 줄] vi에디터에서 esc로 표준모드 접근 후 원하는 줄번호 입력 후 shift+g하게 되면 해당 행으로 찾아감
export JAVA_HOME=/opt/apps/jdk8/ #주석 해제

```
# The java implementation to use. By default, this environment  
# variable is REQUIRED on ALL platforms except OS X!  
export JAVA_HOME=/opt/apps/jdk8/
```

[211번째 줄]
export HADOOP_PID_DIR=/pids #주석 해제

```
# Where pid files are stored. /tmp by default.  
export HADOOP_PID_DIR=/pids
```

모든 설정 후
cd ~ # /home/hadoop 경로로 돌아감

15-4.hadoop 파일 재압축 및 배포

Action server: server01

User: hadoop

Pwd: /home/hadoop

```
ls
```

```
tar cvfz hadoop.tar.gz hadoop-3.1.0 #재압축
```

```
[hadoop@server01 ~]$ ls  
hadoop-3.1.0  hadoop-3.1.0.tar.gz  hadoop.tar.gz  
[hadoop@server01 ~]$
```

```
scp hadoop.tar.gz hadoop@server02:/home/hadoop #배포
```

```
scp hadoop.tar.gz hadoop@server03:/home/hadoop
```

15-5.hadoop 재압축 파일 해제

Action server: server02, server03

User: hadoop

Pwd: /home/hadoop

```
ls
```

```
tar xvfz hadoop.tar.gz #server02, server03에서 압축 풀기
```

16.root계정에서 hadoop 환경변수 설정

16-1.root계정으로 이동

Action server: server02, server03

User: hadoop

Pwd: /home/hadoop

```
exit
```

16-2.hadoop 환경변수 적용

Action server: all

User: root

Pwd: /root

```
vi /etc/profile.d/hadoop.sh #하둡 환경변수 수정
```

```
export HADOOP_HOME=/home/hadoop/hadoop-3.1.0
```

```
export PATH=$PATH:$HADOOP_HOME/bin
```

```
export PATH=$PATH:$HADOOP_HOME/sbin
```

```
source /etc/profile.d/hadoop.sh #변경사항 적용
```

```
export HADOOP_HOME=/home/hadoop/hadoop-3.1.0
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
```


16-3.hadoop 실행을 위해 경로 이동

Action server: all

User: root

Pwd: /root

```
su - hadoop #hadoop 계정으로 접속
```

```
cd hadoop-3.1.0 #경로 이동
```

17.hadoop cluster 실행

17-1.hadoop cluster 실행

Action server: all

User: hadoop

Pwd: /home/hadoop/hadoop-3.1.0

- 차례대로 실행
- format 명령은 초기 실행만, 두 번째 실행부터는 생략, 한 번 더 실행 시 오류 발생 가능성
- 각 명령어마다 실행하는 서버가 다름 ()참조!
- format명령 시 Y/N나오면 대문자 Y로 쓸 것, 대소문자 구별함

```
./bin/hdfs zkfc -formatZK #(*최초 1회, server01)
```

```
./bin/hdfs --daemon start journalnode #(*server01, server02, server03, 저널노드 실행)
```

```
./bin/hdfs namenode -format NNHA #(*최초 1회, server01  
네임노드 초기화, 연결 안 되는 경우 방화벽 해제 확인)
```

```
./bin/hdfs --daemon start namenode #(*server01, 액티브 네임노드 실행)
```

```
./bin/hdfs --daemon start zkfc #(server01, 주키퍼 장애 컨트롤러 실행)
```

```
./sbin/hadoop-daemons.sh start datanode #(server01, 전체 데이터 노드 실행 daemons s 꼭 붙이기)
```

```
./bin/hdfs namenode -bootstrapStandby #(server02, 스탠바이 네임노드 포맷  
액티브 네임노드의 메타데이터를 스탠바이 네임노드로 복사)
```

```
./bin/hdfs --daemon start namenode #(server02, server02에서 네임노드(스탠바이 네임노드)를 실행)
```

```
./bin/hdfs --daemon start zkfc #(server02, 스탠바이 네임노드용 주키퍼 장애 컨트롤러 실행)
```

```
./sbin/start-yarn.sh #(server01, yarn cluster 실행)
```

17-2.hadoop cluster 실행 서버 확인

Action server: all

User: hadoop

Pwd: /home/hadoop/hadoop-3.1.0

jps

[server01]

```
[hadoop@server01 hadoop-3.1.0]$ jps
24784 NameNode
24643 JournalNode
25043 DataNode
25507 NodeManager
25717 WebAppProxyServer
24886 DFSZKFailoverController
25752 Jps
25210 ResourceManager
```

[server02]

```
[hadoop@server02 hadoop-3.1.0]$ jps
24595 DataNode
24756 NameNode
24932 NodeManager
24857 DFSZKFailoverController
25034 Jps
24510 JournalNode
```

[server03]

```
[hadoop@server03 hadoop-3.1.0]$ jps
24585 JournalNode
24876 Jps
24670 DataNode
24782 NodeManager
```

17-3.액티브 스탠바이 노드 확인

Action server: server01

User: hadoop

Pwd: /home/hadoop/hadoop-3.1.0

```
./bin/hdfs haadmin -getServiceState server01 #active
```

```
./bin/hdfs haadmin -getServiceState server02 #standby
```

```
[hadoop@server01 hadoop-3.1.0]$ ./bin/hdfs haadmin -getServiceState server01  
active
```

```
[hadoop@server01 hadoop-3.1.0]$ ./bin/hdfs haadmin -getServiceState server02  
standby
```

```
[hadoop@server01 hadoop-3.1.0]$
```

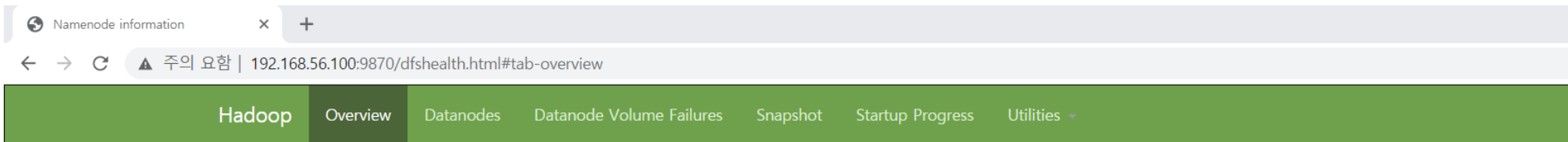
17-4.사이트 접속

Action server: all

User: hadoop

Pwd: /home/hadoop/hadoop-3.1.0

server01(active)은 활성화 되어있고 서버 접속 가능
[server01 ip:9870] *본 가이드는 192.168.56.100:9870



Overview 'server01:8020' (active)

Namespace:	NNHA
Namenode ID:	server01
Started:	Fri Jan 29 16:28:00 +0900 2021
Version:	3.1.0, r16b70619a24cdcf5d3b0fcf4b58ca77238ccbe6d
Compiled:	Fri Mar 30 09:00:00 +0900 2018 by centos from branch-3.1.0
Cluster ID:	CID-90819b93-2862-4c62-b2ef-b5dff521aca0
Block Pool ID:	BP-428711540-192.168.56.100-1611905262972

server02(standby)은 비활성화지만 서버 접속 가능
[server02 ip:9870] *본 가이드는 192.168.56.101:9870
액티브 노드에 문제가 발생하면 스탠바이 노드가 작업을 이어받아 시스템 동작에 문제가 없다 (고가용성:HA)

Namenode information

×

+

← → ↺ ⚠ 주의 요약 | 192.168.56.101:9870/dfshealth.html#tab-overview

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities ▾

Overview 'server02:8020' (standby)

Namespace:	NNHA
Namenode ID:	server02
Started:	Fri Jan 29 16:28:51 +0900 2021
Version:	3.1.0, r16b70619a24cdcf5d3b0fcf4b58ca77238ccbe6d
Compiled:	Fri Mar 30 09:00:00 +0900 2018 by centos from branch-3.1.0
Cluster ID:	CID-90819b93-2862-4c62-b2ef-b5dff521aca0
Block Pool ID:	BP-428711540-192.168.56.100-1611905262972

18.HDFS 확인

Action server: server01

User: hadoop

Pwd: /home/hadoop

cd ~ #/home/hadoop 경로 이동

hdfs dfs -ls / #hdfs 상의 루트 디렉토리 조회

hdfs dfs -mkdir /user #/user 디렉토리 생성

hdfs dfs -mkdir /user/hadoop #/user 디렉토리 하위에 /hadoop 디렉토리 생성

hdfs dfs -mkdir /user/hadoop/conf #/user/hadoop 디렉토리 하위에 /conf 디렉토리 생성

hdfs dfs -put /home/hadoop/hadoop-3.1.0/etc/hadoop/hadoop-env.sh /user/hadoop/conf/
#hadoop-env.sh 파일을 HDFS의 /user/hadoop/conf 디렉토리로 이동

hdfs dfs -ls /user/hadoop/conf #파일 이동 확인

아래와 같이 나온다면 성공

```
[hadoop@server01 ~]$ hdfs dfs -ls /user/hadoop/conf
Found 1 items
-rw-r--r--  3 hadoop supergroup    16393 2021-01-29 02:43 /user/hadoop/conf/hadoop-env.sh
[hadoop@server01 ~]$
```

19. 맵리듀스 확인

Action server: server01

User: hadoop

Pwd: /home/hadoop

wordcount 실행

```
yarn jar /home/hadoop/hadoop-3.1.0/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.0.jar  
wordcount conf output
```

hdfs dfs -ls /user/hadoop/output/ #wordcount 결과 파일 생성 조회

성공 시 아래와 같은 결과

```
[hadoop@server01 ~]$ hdfs dfs -ls /user/hadoop/output/  
Found 2 items  
-rw-r--r--    3 hadoop supergroup          0 2021-01-29 02:48 /user/hadoop/output/_  
SUCCESS  
-rw-r--r--    3 hadoop supergroup    10089 2021-01-29 02:48 /user/hadoop/output/p  
art-r-00000  
[hadoop@server01 ~]$
```

19-1.워드카운트 확인

Action server: server01

User: hadoop

Pwd: /home/hadoop

hdfs dfs -cat /user/hadoop/output/part-r-00000 #wordcount 결과 확인 (파일의 단어 갯수)

```
used      14
user       2
users      4
users,     1
uses       4
using      3
utilize    1
utilized.  1
utilizing  1
value      1
values     1
var:       1
variable   4
variable.  3
variables  2
variables. 1
vast       1
versions.  1
very       1
via        4
vs         1
want       3
wants      2
way        1
when       13
where      1
which      3
who        1
why....    1
wildcards  1
will       29
with       6
work       1
workers.sh, 1
writing    1
writing,    1
xxx-env.sh. 1
yarn       1
yet/still,  1
you        2
{YARN_xyz|HDFS_xyz} 1
{yarn-env.sh|hdfs-env.sh} 1
[hadoop@server01 ~]$
```

워드카운트까지 출력되었다면 구축 성공!

20.hadoop cluster 종료

Action server: all

User: hadoop

Pwd: /home/hadoop/hadoop-3.1.0

- 종료는 실행의 역순으로
- 마찬가지로 명령어마다 실행할 서버가 다름 ()참고

```
cd hadoop-3.1.0 #경로 이동
```

```
./sbin/stop-yarn.sh #(server01, 안 클러스터 종료)
```

```
./bin/hdfs --daemon stop zkfc #(server02, 스탠바이 네임노드용 주키퍼 장애 컨트롤러 종료)
```

```
./bin/hdfs --daemon stop namenode #(server02, 네임노드(스탠바이 네임노드) 종료)
```

```
./sbin/hadoop-daemons.sh stop datanode #(server01, 전체 데이터 노드 종료 daemons s 꼭 붙이기)
```

```
./bin/hdfs --daemon stop zkfc #(server01, 액티브 네임노드용 주키퍼 장애 컨트롤러 종료)
```

```
./bin/hdfs --daemon stop namenode #(server01, 네임노드(액티브 네임노드) 종료)
```

```
./bin/hdfs --daemon stop journalnode #(server01, server02, server03, 각 저널노드 종료)
```

20-1.hadoop cluster 종료 확인

Action server: all

User: hadoop

Pwd: /home/hadoop

jps

정상적으로 종료되었다면 모든 서버에서 jps만 출력됨

```
[hadoop@server01 hadoop-3.1.0]$ jps
27203 Jps
[hadoop@server01 hadoop-3.1.0]$
```

```
[hadoop@server02 hadoop-3.1.0]$ jps
25761 Jps
[hadoop@server02 hadoop-3.1.0]$
```

```
[hadoop@server03 hadoop-3.1.0]$ jps
25290 Jps
[hadoop@server03 hadoop-3.1.0]$
```

21.zookeeper cluster 종료

21-1.주키퍼 계정으로 전환 후 경로 이동

Action server: all

User: hadoop

Pwd: /home/hadoop

```
exit
```

```
su - zookeeper
```

```
cd zookeeper-3.4.10
```

21-2.zookeeper cluster 서버 종료

Action server: all

User: zookeeper

Pwd: /home/zookeeper/zookeeper-3.4.10

```
./bin/zkServer.sh stop #주키퍼 종료
```

```
jps #마찬가지로 jps만 출력되면 정상 종료
```

```
[zookeeper@server01 zookeeper-3.4.10]$ jps
27262 Jps
[zookeeper@server01 zookeeper-3.4.10]$ █
```

```
[zookeeper@server02 zookeeper-3.4.10]$ jps
25808 Jps
[zookeeper@server02 zookeeper-3.4.10]$ █
```

```
[zookeeper@server03 zookeeper-3.4.10]$ jps
25334 Jps
[zookeeper@server03 zookeeper-3.4.10]$ █
```


ERRORS

DFSZFailoverController 서버가 실행되지 않을 때

- zookeeper 서버가 실행 되어있는지 확인

jps 서버가 실행되지 않을 때

- java 설정 확인, 환경 변수 등

권한 거부 에러

- 각 계정별 ssh 공유 확인, 방화벽 해제 확인

namenode가 실행되지 않을 때

- hdfs-site.xml 확인

datanode가 실행되지 않을 때

- hdfs-site.xml 확인

resourcemanager, nodemanager가 실행되지 않을 때

- yarn-site.xml 확인

journalnode가 실행되지 않을 때

- hdfs-site.xml 확인