

# 리눅스 시스템 관리

---

01. 리눅스 소개

**02. 리눅스 서버 구축**

03. 리눅스 기본 관리

04. vi 에디터

05. 리눅스 네트워크 관리

06. 리눅스 내부구조

(프로세스,메모리,파일시스템)

## 2.1 리눅스 서버 배포판

### 우분투(ubuntu)

- 데비안 계열의 배포판(캐노니컬, 우분투 재단이 개발, 배포, 유지보수)
- 다른 배포판에 비해 편리하고 사용자 친화적인 인터페이스로 가장 많은 사용층을 가진 배포판이다.
- 데스크 탑에 중점을 둔 개발로 일반인의 리눅스에 대한 쉬운 접근을 가능하게 하였고 실제로 개인용 데스크 탑과 노트북에서 가장 인기 있는 리눅스 배포판이다.
- 서버 배포판으로 나누어져 서버 플랫폼을 지원하고 있다.
- <http://www.ubuntu.com/download/server> 에서 다운로드가 가능하고 최신 버전은 14.04 LTS(안정 버전)이다.



### CentOS

- 리눅스 배포판 중 가장 높은 점유율을 자랑했던 RedHat 계열의 리눅스 배포판이다.
- 레드햇 엔터프라이즈 리눅스(RHEL, 상용)와 완벽하게 호환되는 무료 기업용 컴퓨팅 플랫폼을 제공할 목적으로 만들어진 배포판이다.
- 안정성 부분에서 다른 배포판에 비해 가장 신뢰할 수 있으며 관련 문서와 서버 패키지도 역사만큼 많이 존재한다.
- [http://ftp.daumkakao.com/centos/7/isos/x86\\_64/](http://ftp.daumkakao.com/centos/7/isos/x86_64/) 에서 다운로드 받을 수 있으며 최신 버전은 CentOS7(kernel 3.10.0) 이다.



## 2.2 인스톨 디스크의 파티션

- 일반적으로 리눅스를 설치할 디스크에 파티션을 나눠서 사용한다.
- 특별한 이유가 없는 한 다음 5개 정도의 파티션으로 나누어 설치하게 된다.

파티션	포맷 및 마운트 디렉토리
부트 파티션	ext3, ext4, xfs 파일 시스템으로 포맷하고 /boot 디렉토리로 사용한다.
루트 파티션	ext3, ext4, xfs 파일 시스템으로 포맷하고 / 디렉토리로 사용
swap 파티션	swap 영역으로 포맷
kdump	ext3, ext4, xfs 파일 시스템으로 포맷하고 마운트 하지 않음
데이터용 파티션	애플리케이션 데이터영역으로서 사용

1. 부트 파티션은 커널과 초기 RAM 디스크를 포함, GRUB에 관련된 파일을 저장하기 때문에 100(보통128)MB 정도면 충분하다.
2. /(루트) 파티션은 인스톨 예정의 RPM 패키지와 애플리케이션 소프트웨어를 인스톨 할 수 있는 정도의 사이즈가 필요
3. swap 영역은 RHEL에서 추천하는 다음 계산을 따른다.  
메모리가 2GB 이하 : 물리 메모리의 2배  
메모리가 2GB 이상 : 물리 메모리 + 2GB

## 2.2 인스톨 디스크의 파티션

4. kdump는 물리 메모리 사이즈의 2~3배 정도가 알맞다.

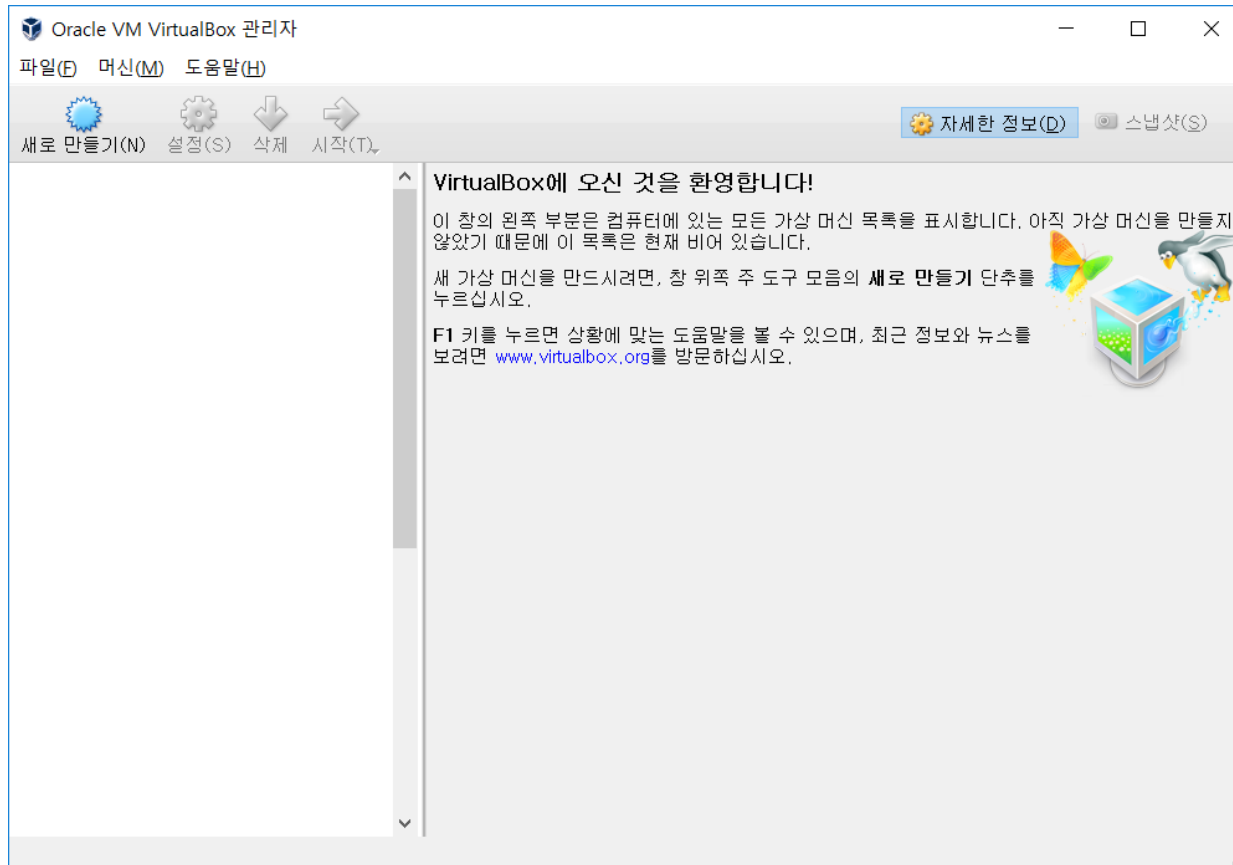
5. 데이터용 파티션은 애플리케이션 데이터 영역으로 운용 중에 데이터가 계속 증가할 영역으로 따로 파티션을 잡아 두는 것이 좋다.

- 디스크 파티션 설정 화면에서는 “Default Layout” 이 아닌 “Custom Layout”를 선택해서 명시적으로 파티션을 구성 하도록 한다.
- LVM(Logical Volume Management)는 사용하지 않는 것이 원칙이다. (설치 시)
- 데이터 양이 일정 한 영역과 데이터가 증가하는 영역으로 나뉘서 관리하는 것이 디스크 관리의 기본이다.

## 2.3 리눅스 인스톨

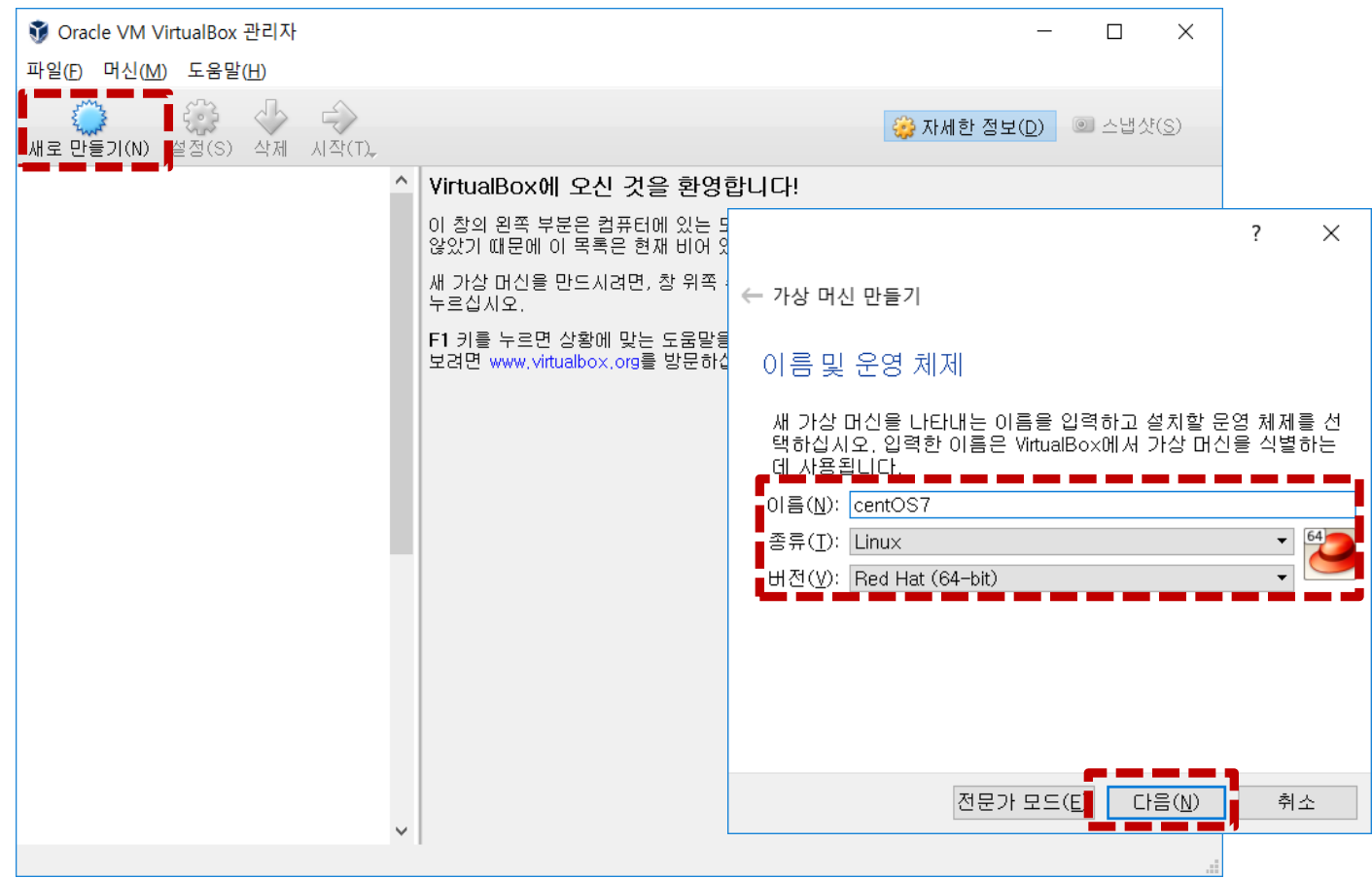
### 2.3.1 Virtual Box

- 실제 서버 장비의 운용 전에 테스트 용도로 사용
- 최근에는 가상 서버를 이용한 상용 호스팅 서비스도 안정화 되어가고 증가되고 있는 추세다.
- Oracle VM VirtualBox 에 CentOS7를 설치 하도록 한다.



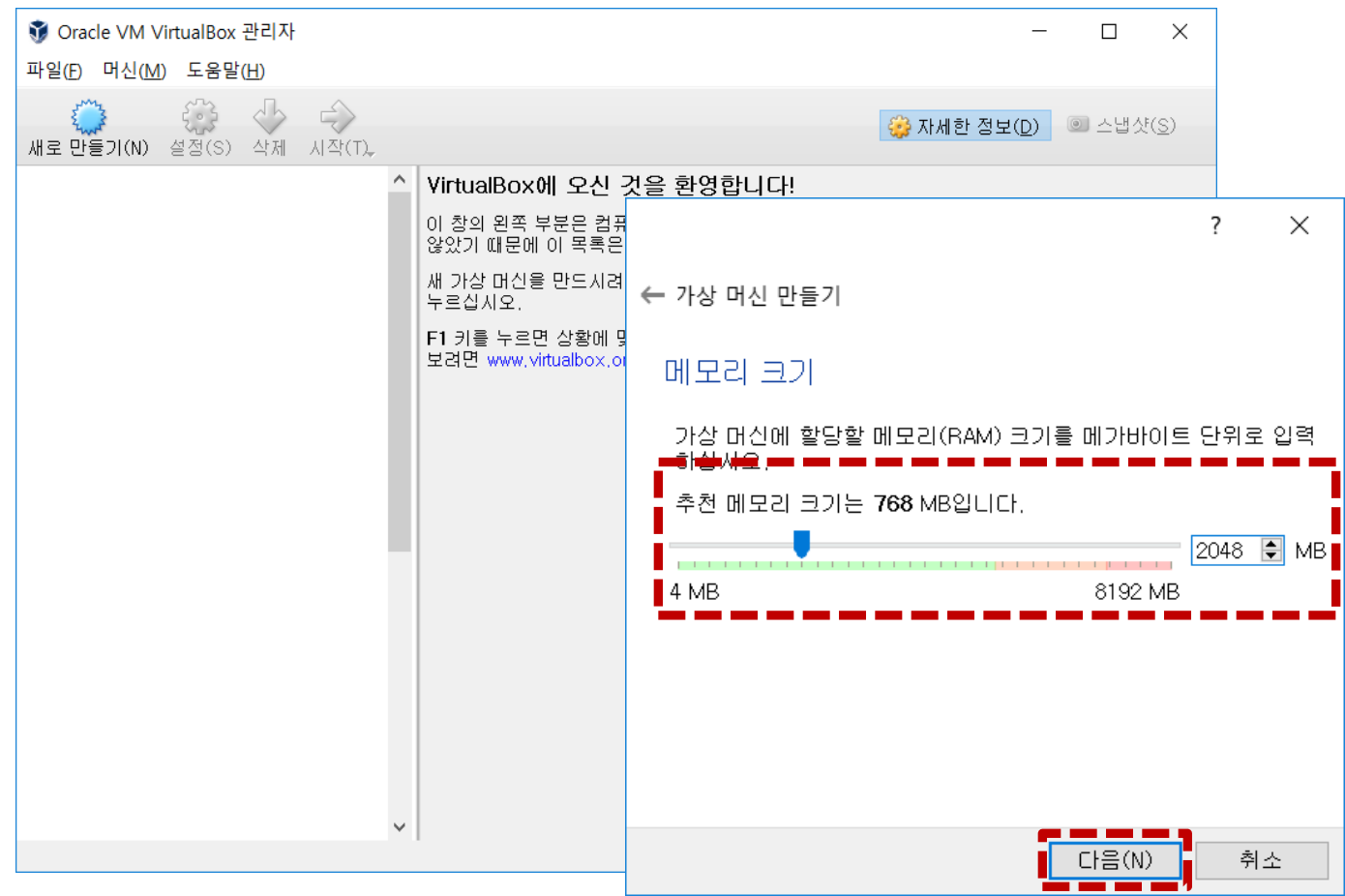
## 2.3 리눅스 인스톨

### 2.3.2 새 가상 머신 만들기



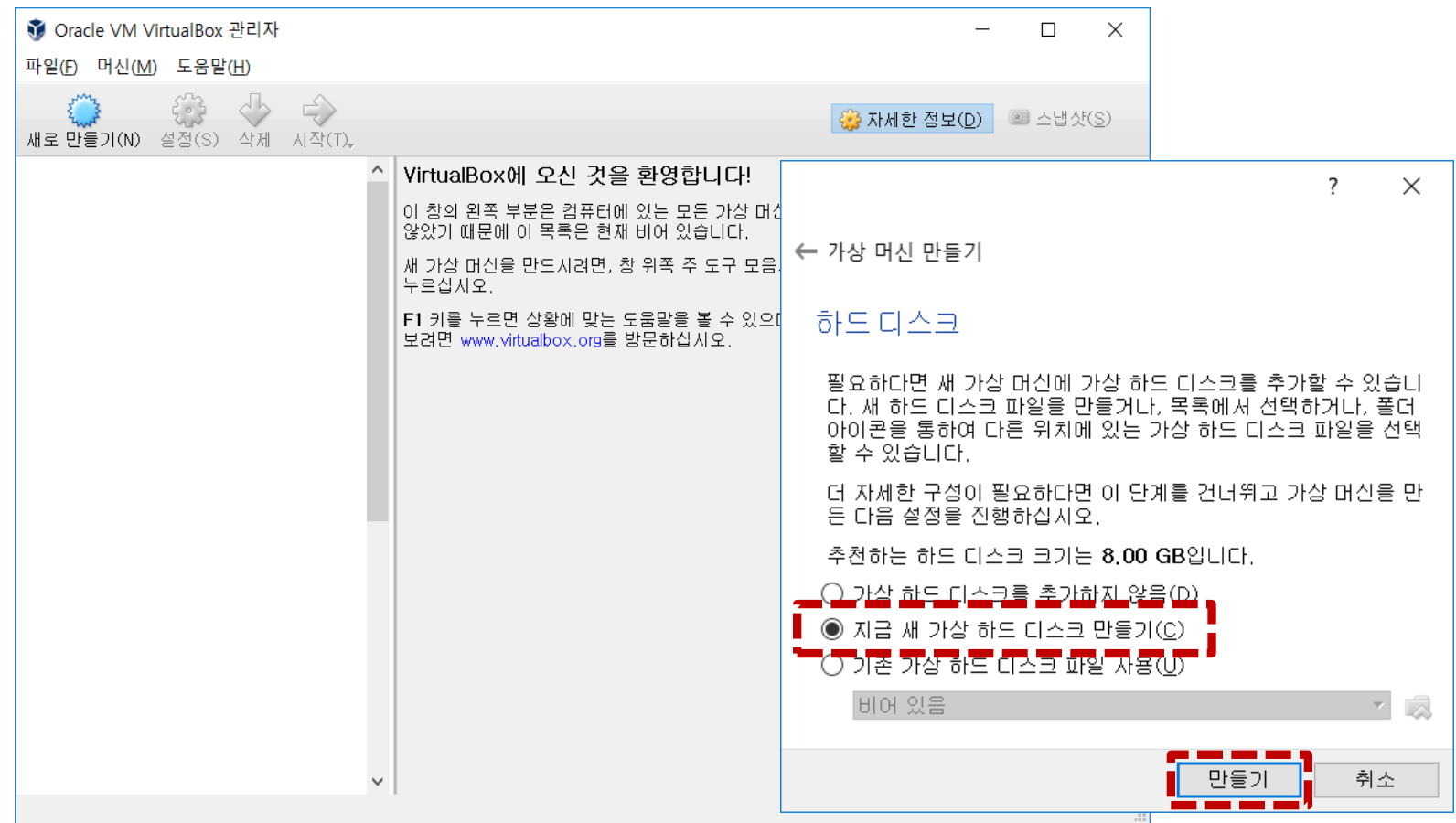
## 2.3 리눅스 인스톨

### 2.3.2 새 가상 머신 만들기



## 2.3 리눅스 인스톨

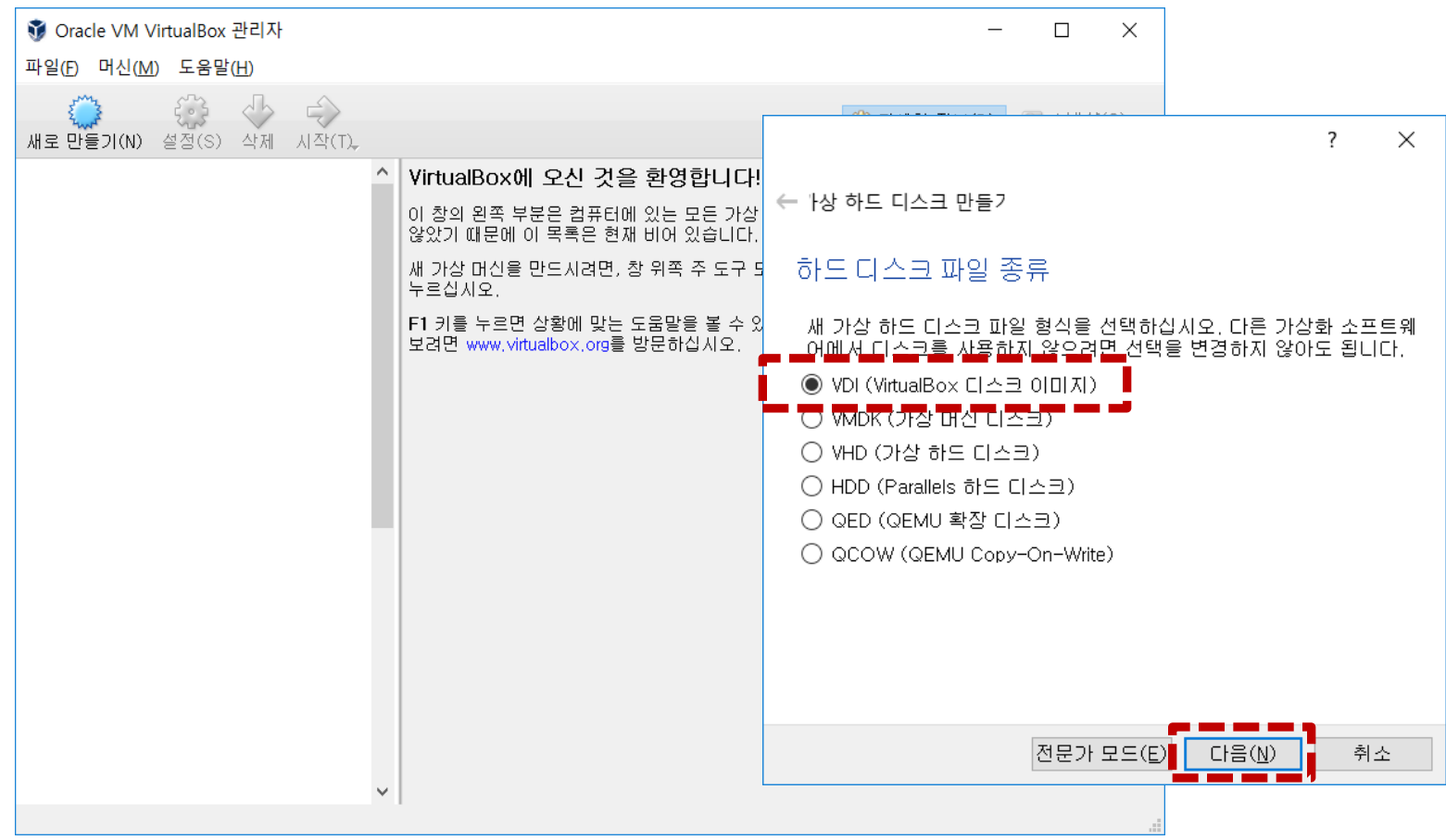
### 2.3.2 새 가상 머신 만들기





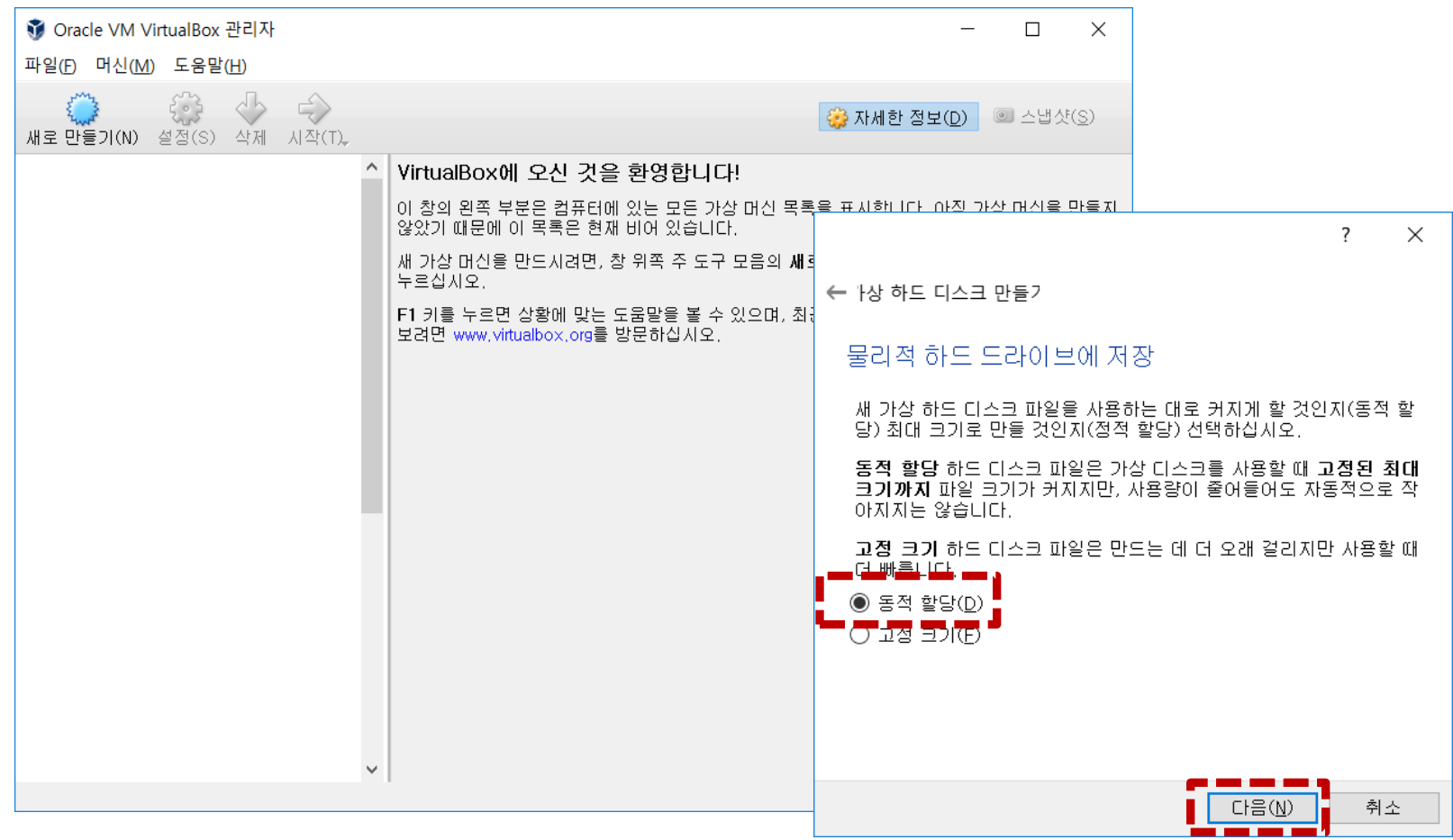
## 2.3 리눅스 인스톨

### 2.3.2 새 가상 머신 만들기



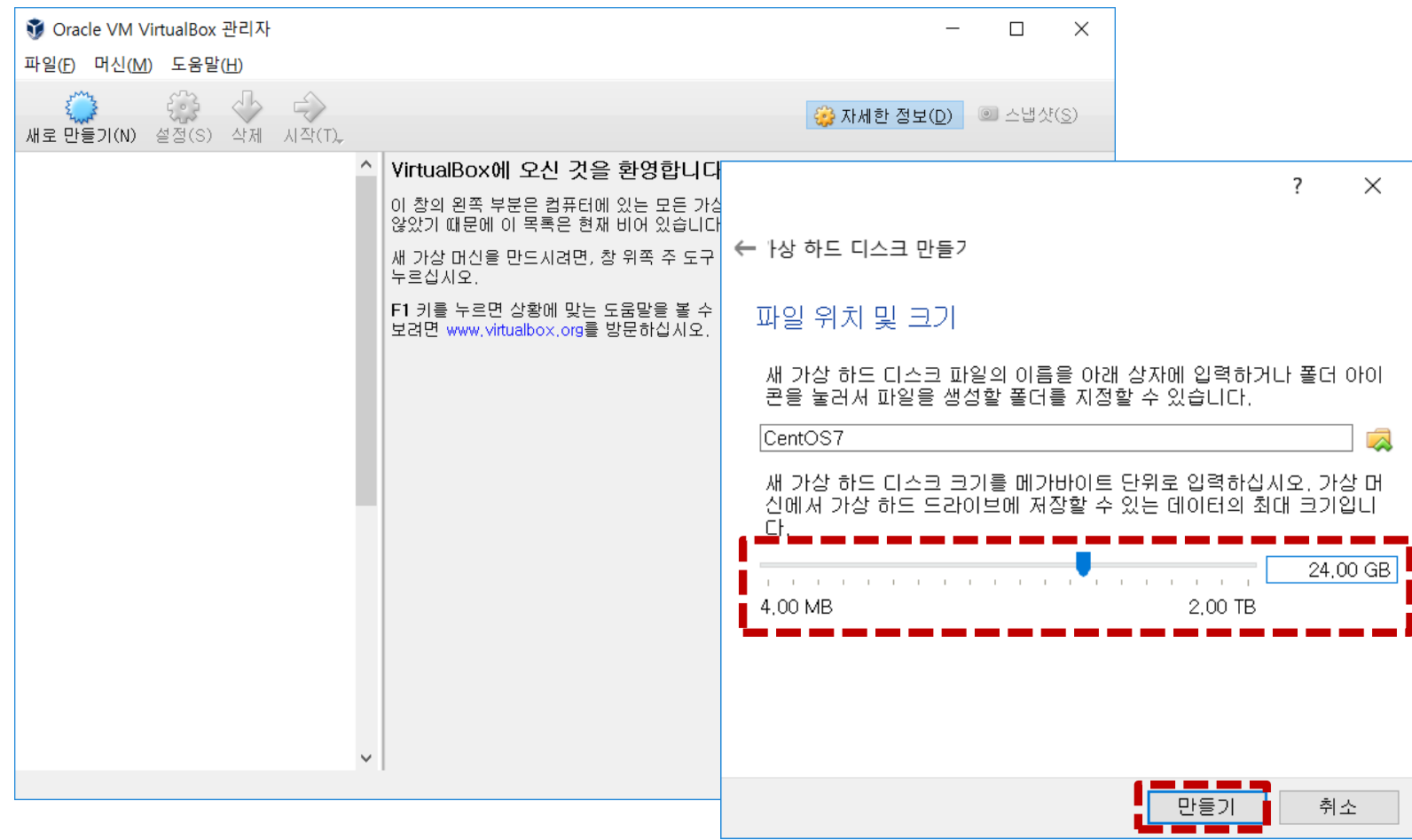
# 2.3 리눅스 인스톨

## 2.3.2 새 가상 머신 만들기



## 2.3 리눅스 인스톨

### 2.3.2 새 가상 머신 만들기



## 2.3 리눅스 인스톨

### 2.3.2 새 가상 머신 만들기

Oracle VM VirtualBox 관리자

파일(F) 머신(M) 도움말(H)

새로 만들기(N)

설정(S)

삭제

시작(T)

64

CentOS7

전원 꺼짐

자세한 정보(D)

스냅샷(S)

일반

이름: CentOS7  
운영 체제: Red Hat (64-bit)

시스템

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

디스플레이

비디오 메모리: 12 MB  
원격 데스크톱 서버: 사용 안함  
비디오 캡처: 사용 안함

저장소

컨트롤러: IDE  
IDE 세컨더리 마스터: [광학 드라이브] 비어 있음  
컨트롤러: SATA  
SATA 포트 0: CentOS7.vdi (일반, 24.00 GB)

오디오

호스트 드라이버: Windows DirectSound  
컨트롤러: ICH AC97

네트워크

어댑터 1: Intel PRO/1000 MT Desktop (NAT)

미리 보기

CentOS7

CentOS7 [실행 중] - Oracle VM VirtualBox

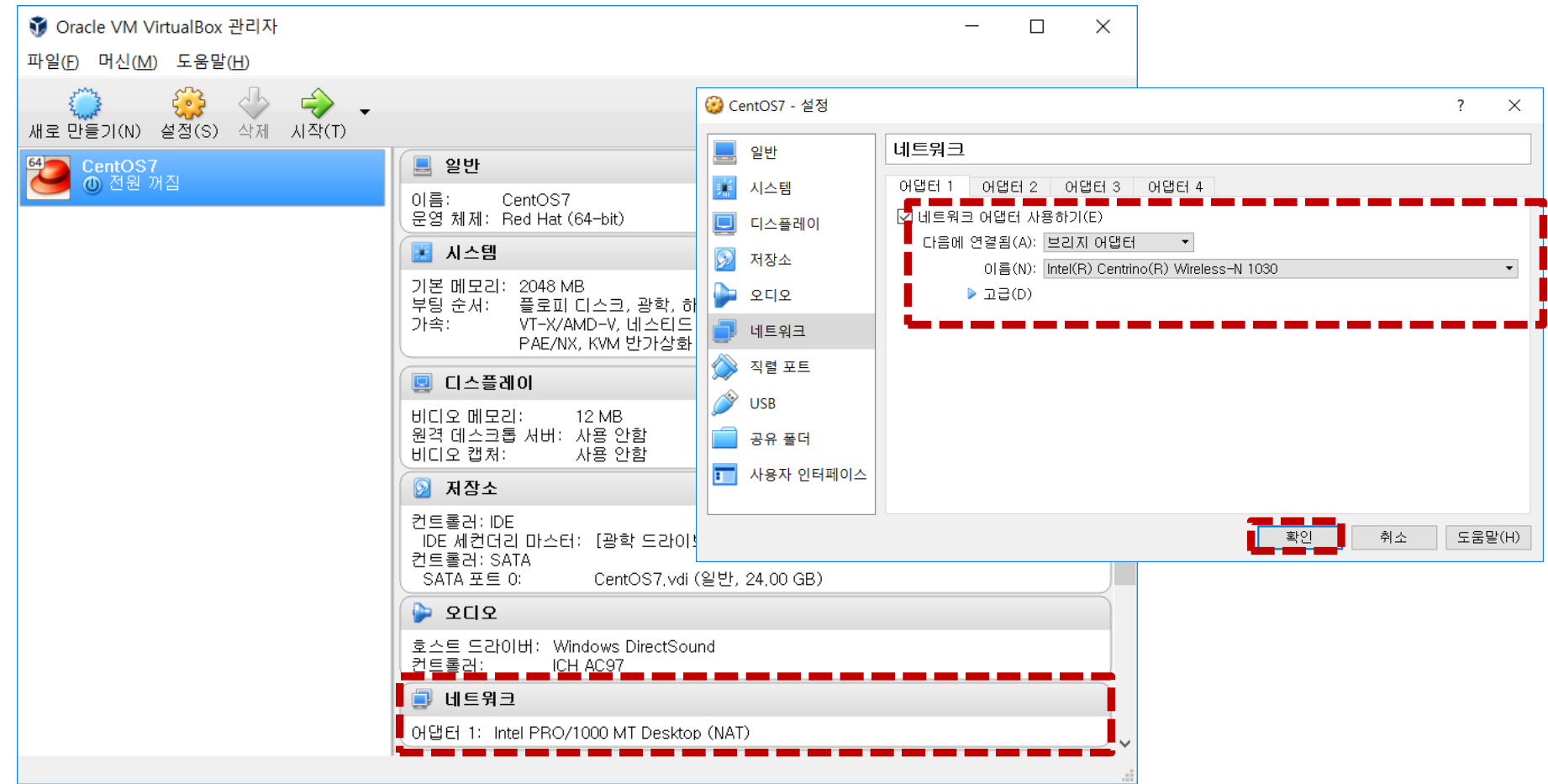
파일 머신 보기 입력 장치 도움말

FATAL: No bootable medium found! System halted.

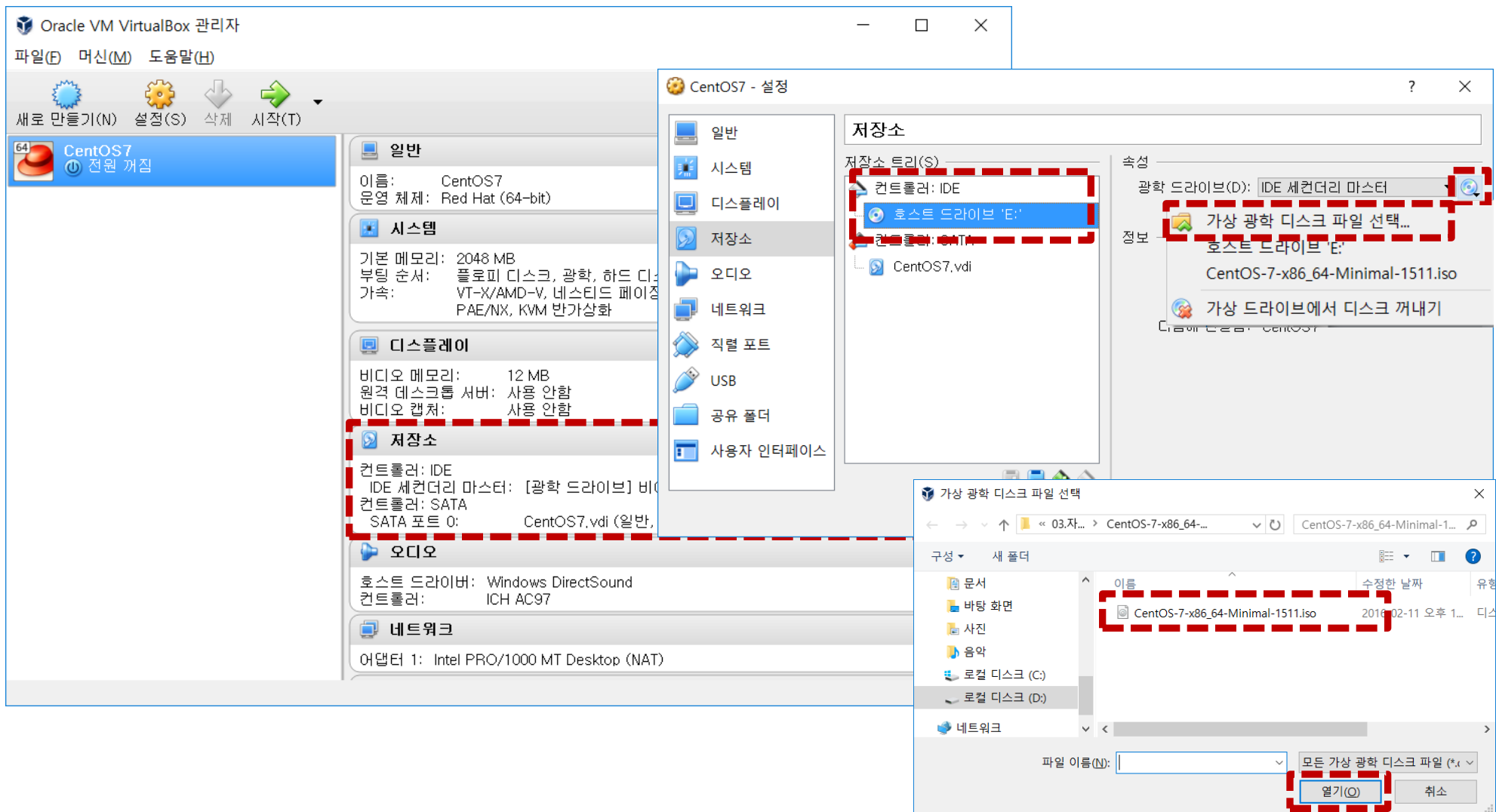
새로 만든 가상 머신에 전원을 켜 보자.

# 2.3 리눅스 인스톨

## 2.3.3 가상 머신 설정 - 네트워크

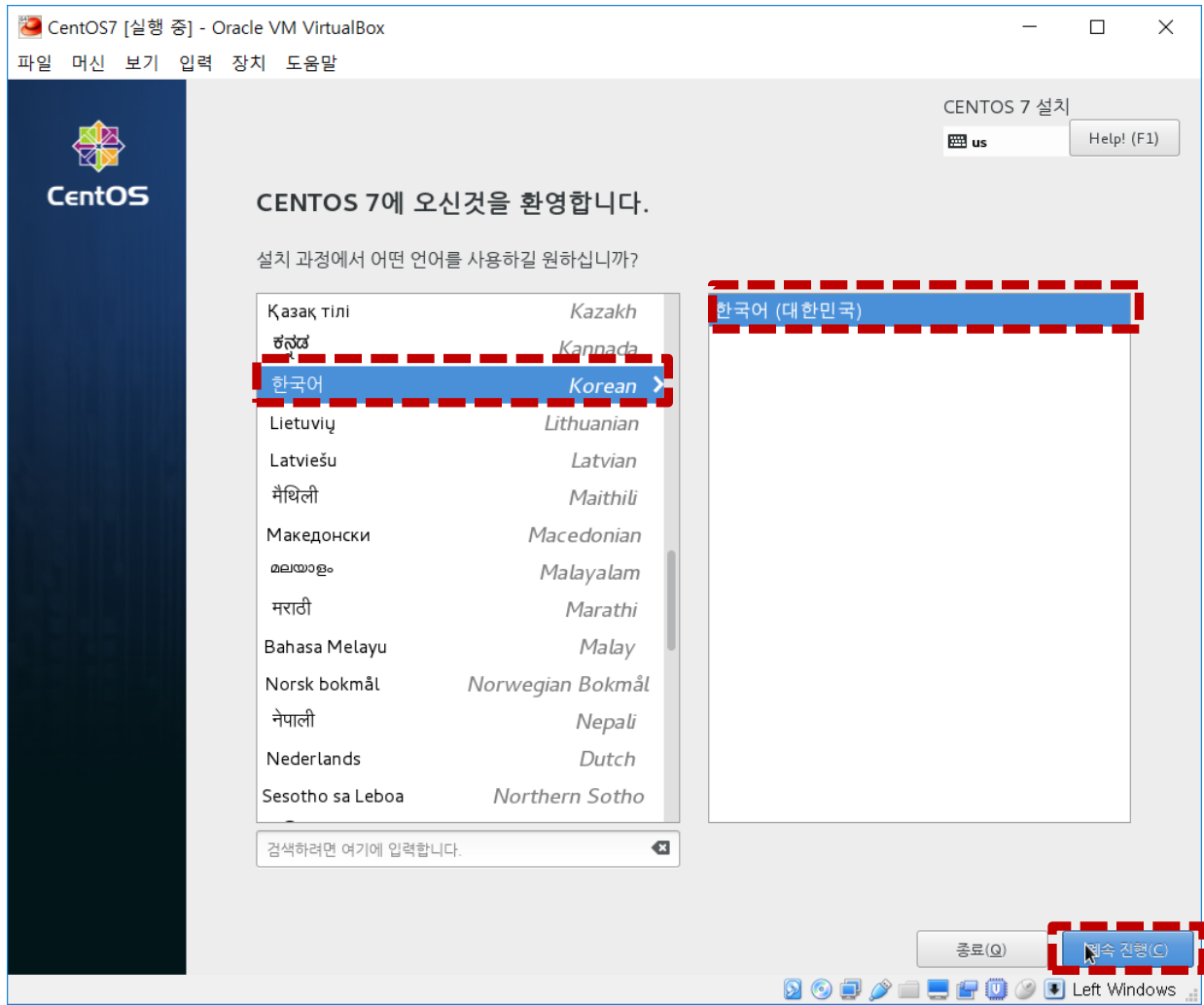


### 2.3.3 가상 머신 설정 - 설치 디스크 이미지 삽입



## 2.3 리눅스 인스톨

### 2.3.4 CentOS 설치 - 1. 설치 과정 언어 선택



## 2.3 리눅스 인스톨

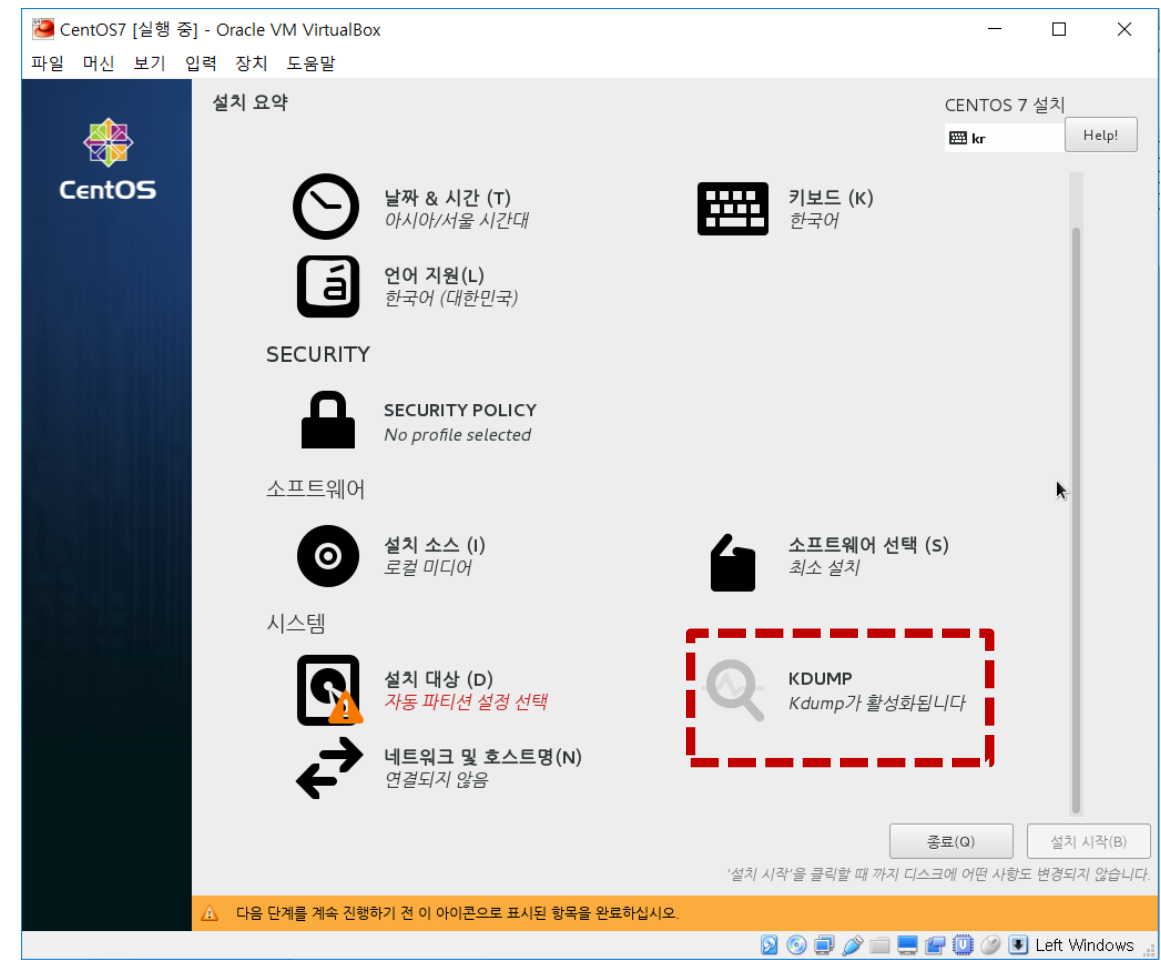
### 2.3.4 CentOS 설치 - 2. 설치 요약 항목 확인





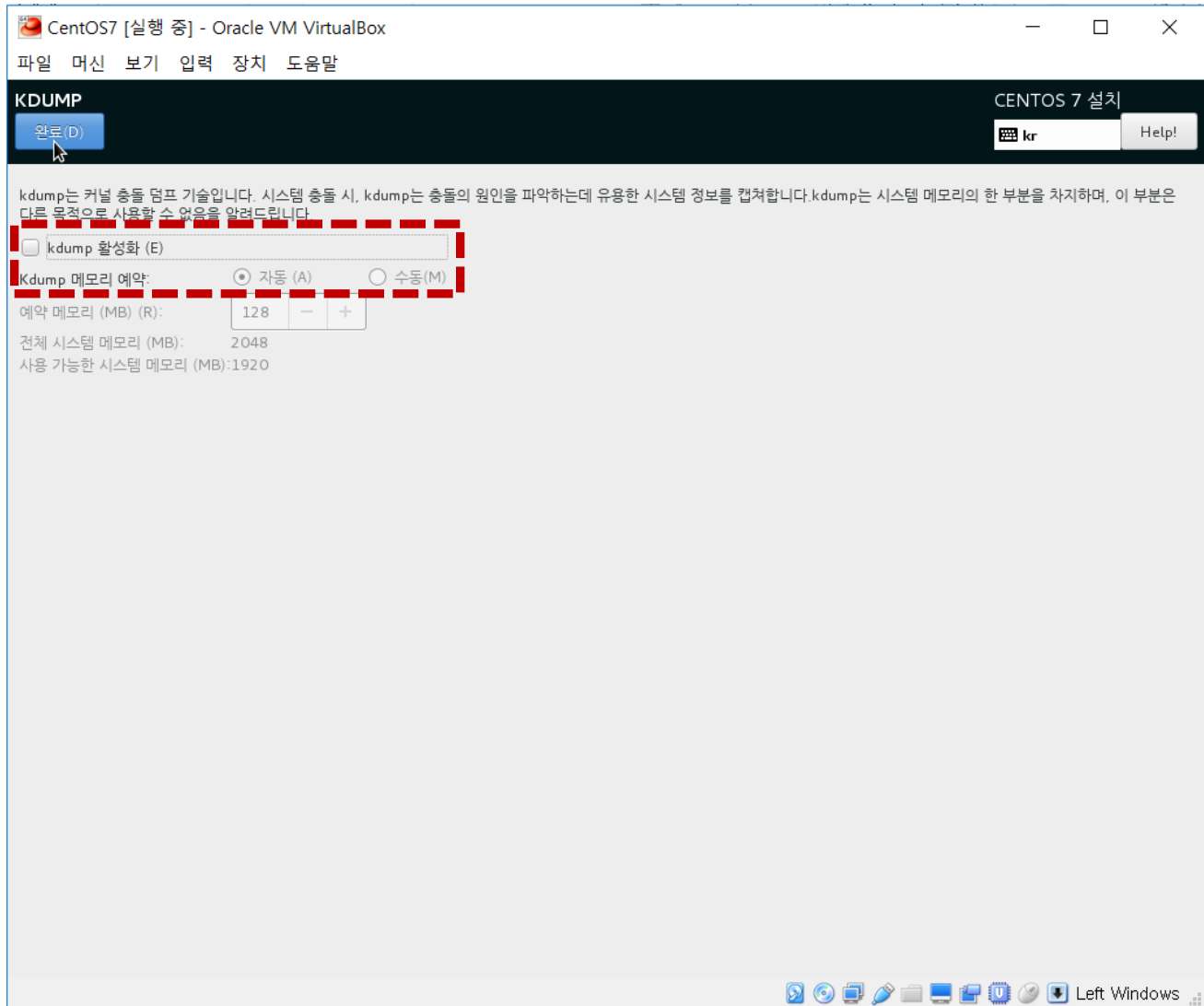
## 2.3 리눅스 인스톨

### 2.3.4 CentOS 설치 - 3. KDUMP 비 활성화 시키기



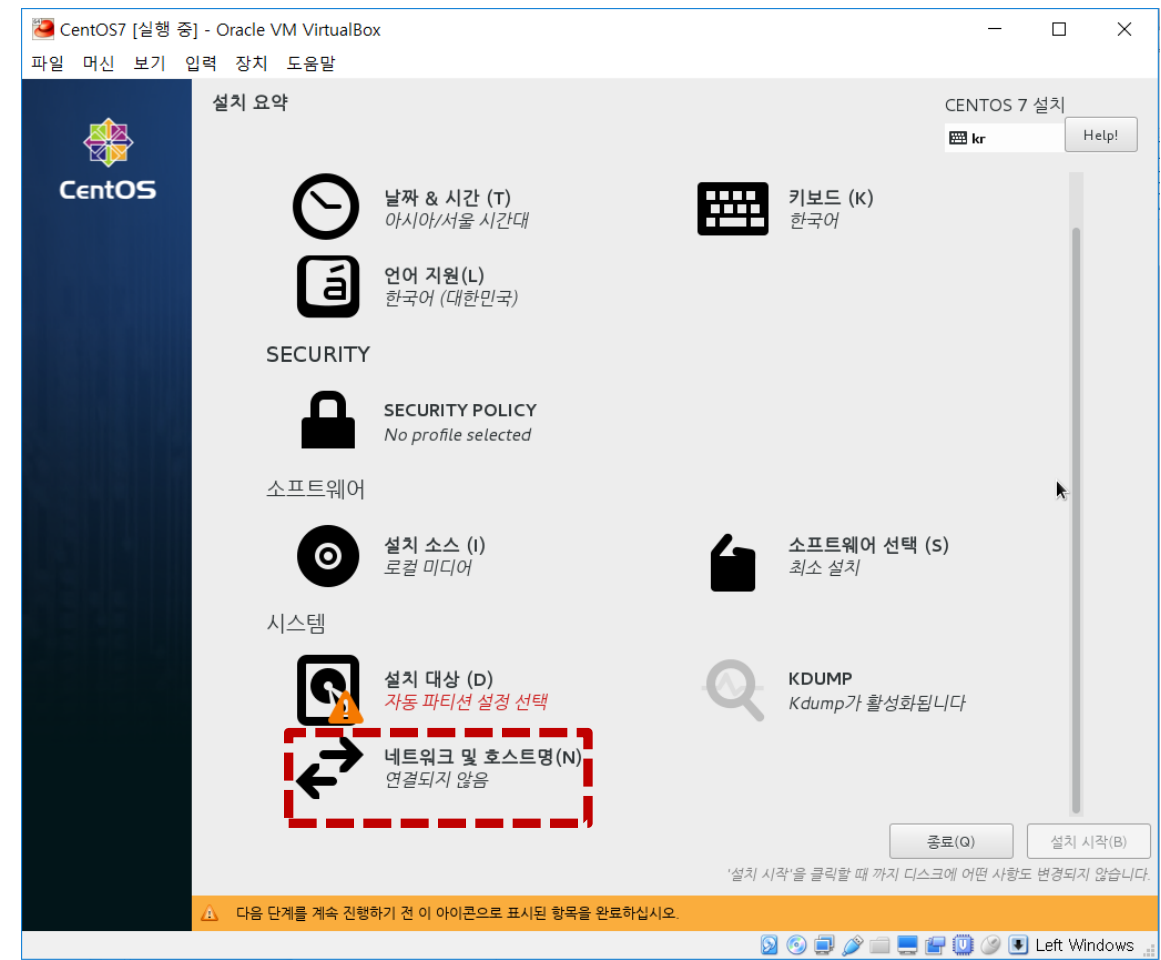
## 2.3 리눅스 인스톨

### 2.3.4 CentOS 설치 - 3. KDUMP 비 활성화 시키기



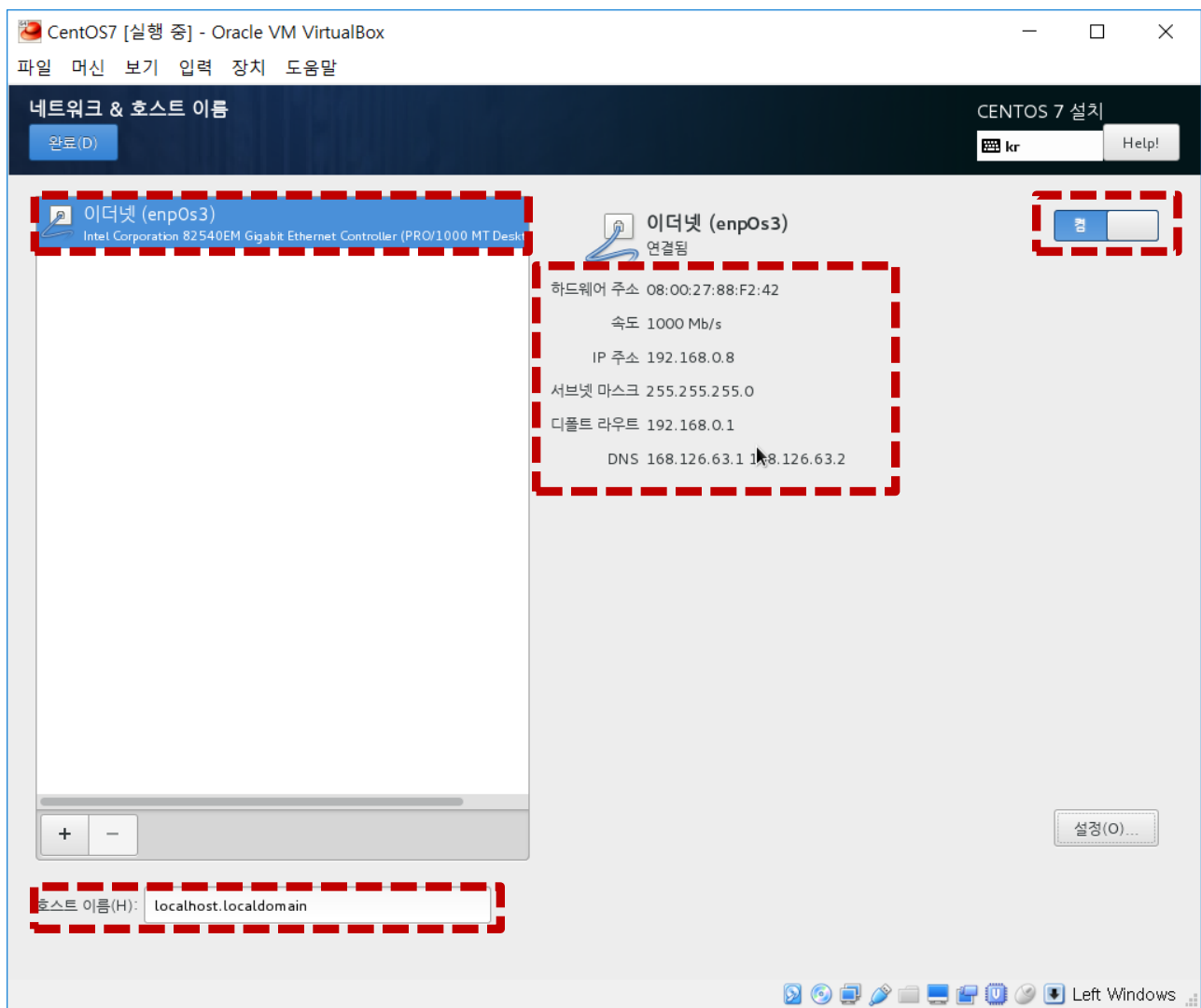
## 2.3 리눅스 인스톨

### 2.3.4 CentOS 설치 - 4. 네트워크 및 호스트명



## 2.3 리눅스 인스톨

### 2.3.4 CentOS 설치 - 4. 네트워크 및 호스트명



## 2.3 리눅스 인스톨

### 2.3.4 CentOS 설치 - 5. 파티션 설정



## 2.3 리눅스 인스톨

### 2.3.4 CentOS 설치 - 5. 파티션 설정

CentOS7 [실행 중] - Oracle VM VirtualBox

파일 머신 보기 입력 장치 도움말

선택 대상

완료(D)

CENTOS 7 설치

kr Help!

장치 선택

설치하려는 장치를 선택합니다. 메인 메뉴의 "설치 시작" 버튼을 클릭하지 않으면 설치는 시작되지 않습니다.

토글 표준 디스크

24 GiB

ATA VBOX HARDDISK

sda / 24 GiB 여유 공간

여기서 선택하지 않은 디스크의 경우 아무것도 실행되지 않습니다.

특수 디스크 및 네트워크 디스크

디스크 추가 (A)...

여기서 선택하지 않은 디스크의 경우 아무것도 실행되지 않습니다.

기타 저장소 옵션

파티션 설정

☐ 파티션을 자동으로 설정합니다. (U)

☒ 파티션을 설정합니다. (I)

☐ 추가 공간을 사용할 수 있게 합니다. (M)

암호화

☐ 데이터를 암호화합니다. (E) 암호는 나중에 설정합니다.

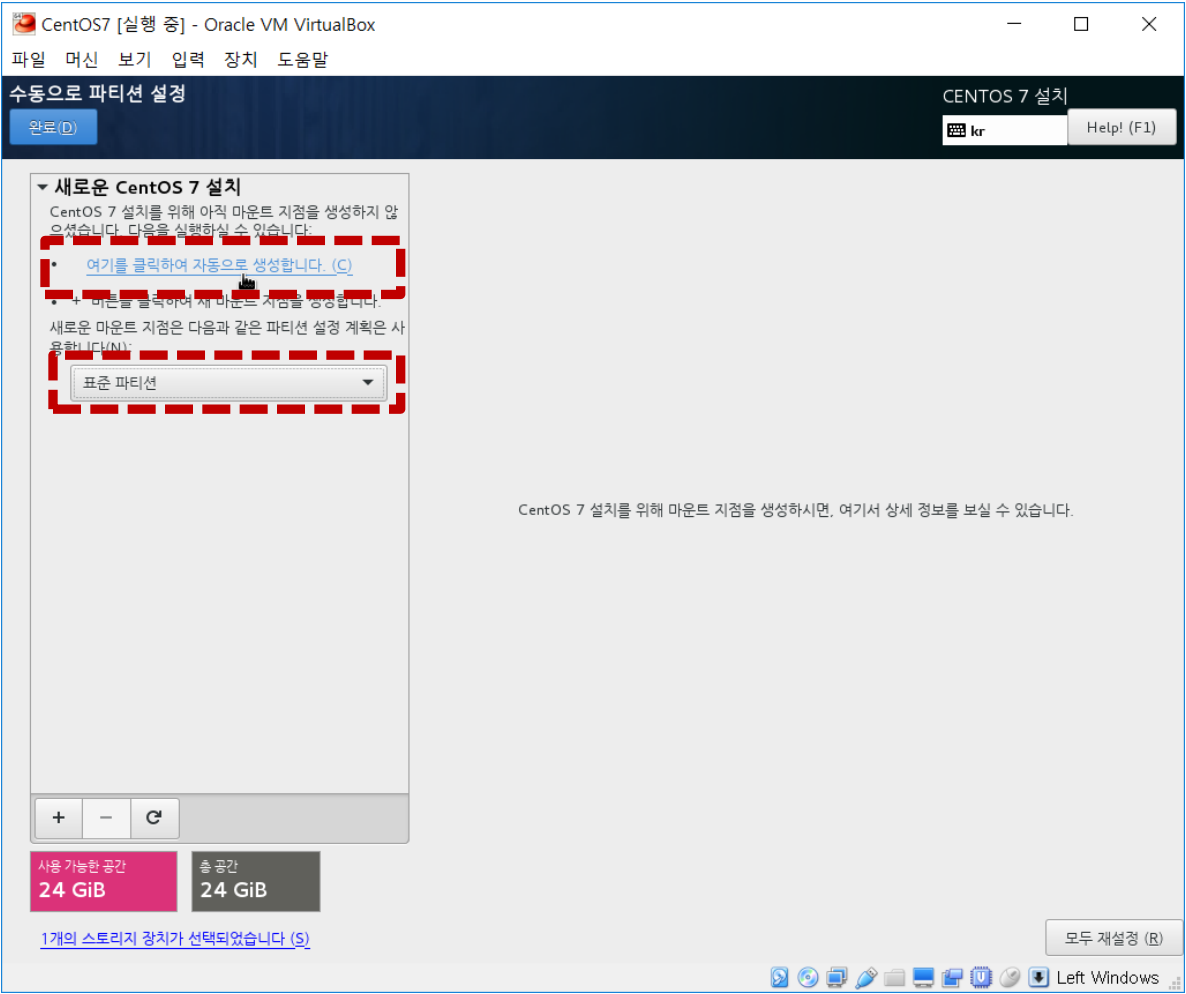
전체 디스크 요약 및 부트 로더 (F)...

1 디스크 선택: 24 GiB 전체 용량: 24 GiB 여유 용량

Left Windows

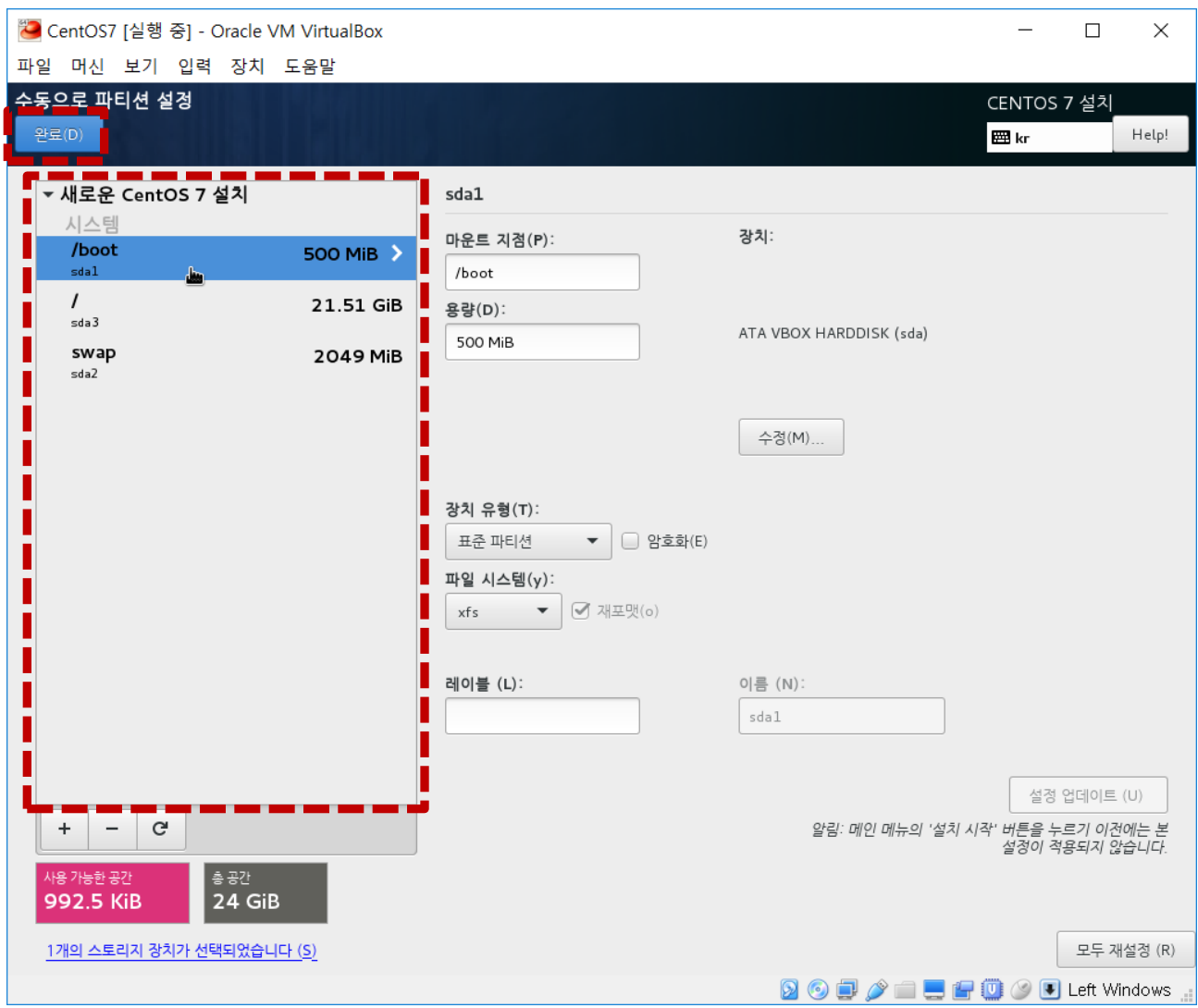
## 2.3 리눅스 인스톨

### 2.3.4 CentOS 설치 - 5. 파티션 설정



# 2.3 리눅스 인스톨

## 2.3.4 CentOS 설치 - 5. 파티션 설정





# 2.3 리눅스 인스톨

## 2.3.4 CentOS 설치 - 5. 파티션 설정

CentOS7 [실행 중] - Oracle VM VirtualBox

파일 머신 보기 입력 장치 도움말

수동으로 파티션 설정

완료(D)

CENTOS 7 설치

kr Help!

새로운 CentOS 7 설치

시스템

/boot 500 MiB >

sda1

/

sda3

swap

sda2

sda1

마운트 지점(P):

장치:

/boot

변경 요약

메인 메뉴로 돌아가 설치를 시작하면 설정이 다음과 같이 적용됩니다:

순서	동작	유형	장치 이름	마운트 지점
1	Destroy Format	Unknown	sda	
2	Create Format	partition table (MSDOS)	sda	
3	Create Device	partition	sda1	
4	Create Device	partition	sda2	
5	Create Format	swap	sda2	
6	Create Device	partition	sda3	
7	Create Format	xfs	sda3	/
8	Create Format	xfs	sda1	/boot

취소하고 사용자 정의 파티션 설정으로 되돌아갑니다 (C) 변경 사항 적용 (A)

설정 업데이트 (U)

알림: 메인 메뉴의 '설치 시작' 버튼을 누르기 이전에는 본 설정이 적용되지 않습니다.

모두 재설정 (R)

사용 가능한 공간 992.5 KiB

총 공간 24 GiB

1개의 스토리지 장치가 선택되었습니다 (S)

Left Windows

## 2.3 리눅스 인스톨

### 2.3.4 CentOS 설치 - 6. 설치 시작



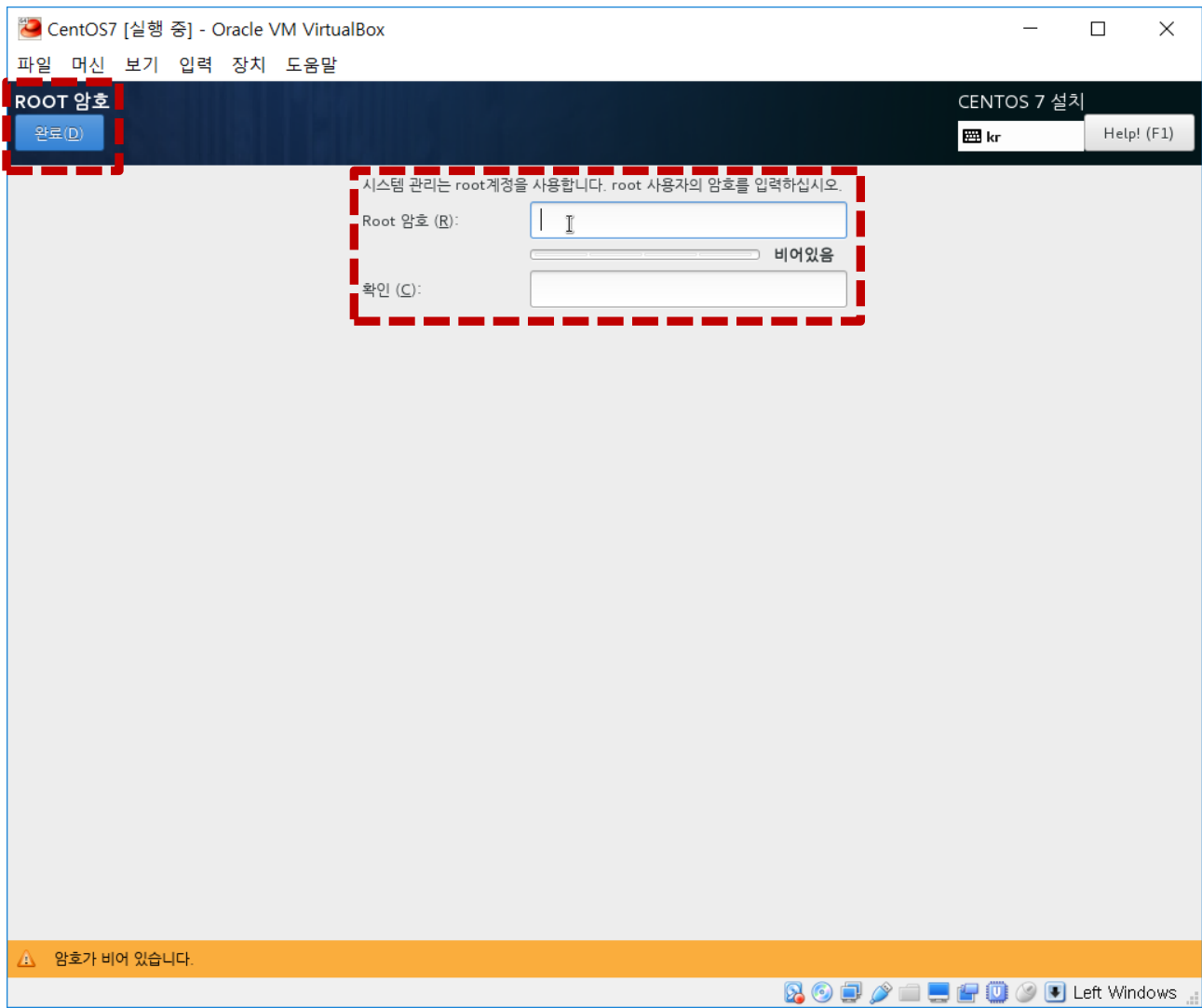
## 2.3 리눅스 인스톨

### 2.3.4 CentOS 설치 - 7. 설치 중 [ root 암호 설정]



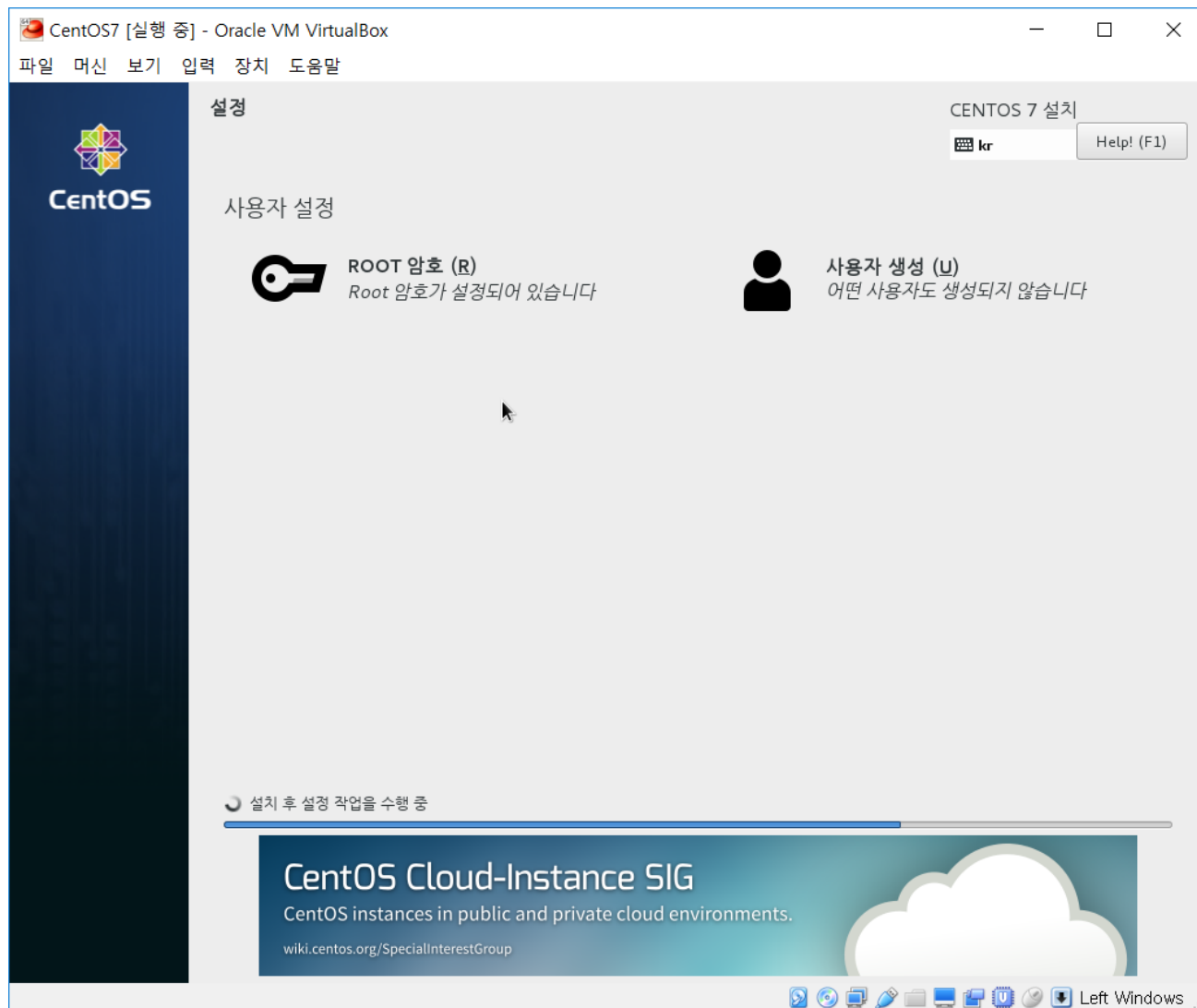
## 2.3 리눅스 인스톨

### 2.3.4 CentOS 설치 - 7. 설치 중 (root 암호 설정)



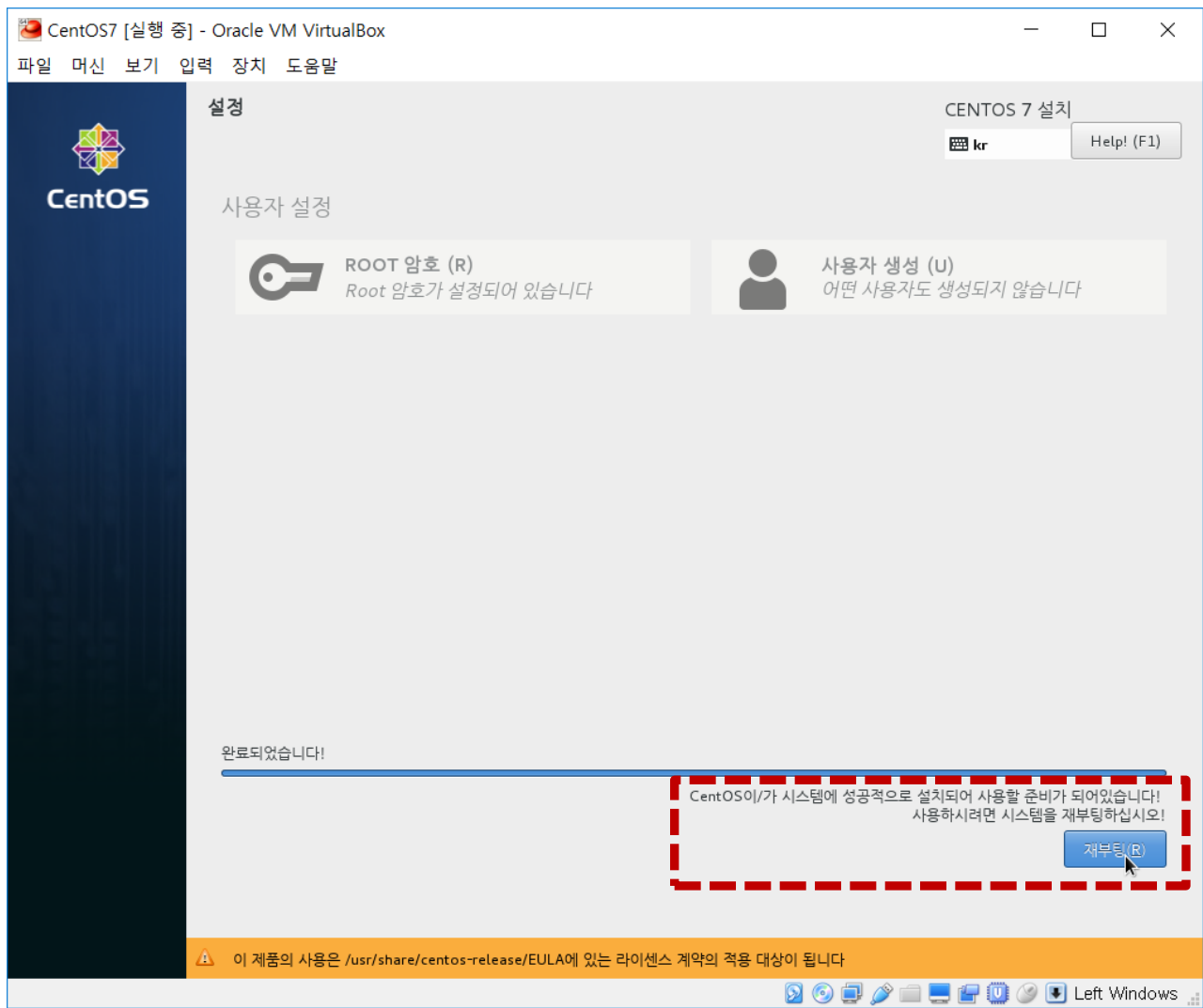
## 2.3 리눅스 인스톨

### 2.3.4 CentOS 설치 - 7. 설치 중



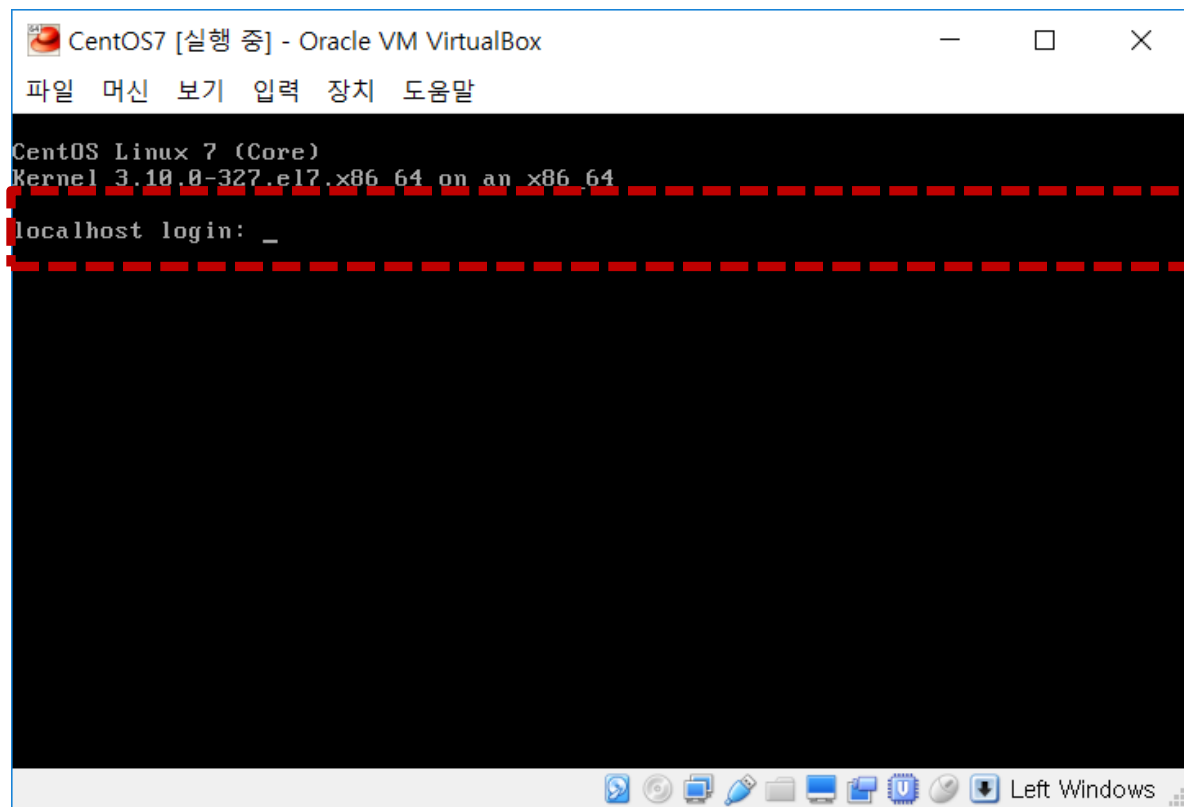
## 2.3 리눅스 인스톨

### 2.3.4 CentOS 설치 - 8. 설치 완료



## 2.3 리눅스 인스톨

### 2.3.4 CentOS 설치 - 9. 재부팅



## 2.4 인스톨 후 기본 설정 작업

### 2.4.1 로그인

로그인 프롬프트에서 계정과 계정 비밀번호를 입력한다.

```
CenOS Linux 7 (Core)  
Kernel 3.10.0-327.el7.x86_64 on an x86_64
```

```
localhost login: _
```

### 2.4.2 로그아웃

```
[root@localhost ~]# exit  
또는  
[root@localhost ~]# logout
```

### 2.4.3 리눅스 버전 확인

```
[root@localhost ~]# uname -a  
Linux localhost.localdomain 3.10.0-327.el7.x86_64 #1 SMP Thu Nov 19 22:10:57 UTC 2015 x86_64  
x86_64 GNU/Linux  
[root@localhost ~]# uname -r  
3.10.0-327.el7.x86_64
```



## 2.4 인스톨 후 기본 설정 작업

### 2.4.4 시스템 종료

#### shutdown

다양한 옵션으로 시스템을 종료할 수 있는 가장 일반적인 명령이다.

```
[root@localhost ~]# shutdown -h now           ( 지금 halt )

[root@localhost ~]# shutdown -h +10           ( 10분 후에 halt )
[root@localhost ~]# shutdown -r 22:10         ( 22:10 에 reboot )
[root@localhost ~]# shutdown -c               ( 진행 중인 shutdown cancel )
[root@localhost ~]# shutdown -h +10 "hurry up" ( 지금 접속 중인 사용자에게 메세지 전송 하고 10분 후에 halt )
```

#### halt

#shutdown -h now 와 동일

```
[root@localhost ~]# halt
```

#### init

런레벨 (시스템 실행 모드, 0~6)를 설정하는 명령으로 0(종료)을 주게 되면 종료된다.

```
[root@localhost ~]# init 0
```

#### systemctl

```
[root@localhost ~]# systemctl poweroff (Halt & Power-Off)
```

```
[root@localhost ~]# systemctl halt (Halt)
```

## 2.4 인스톨 후 기본 설정 작업

### 2.4.5 시스템 재시작

reboot 명령 또는 init을 사용해서 런레벨 6(시스템 재기동)을 주게 되면 재시작을 하게 된다.

```
# reboot
```

(또는)

```
# init 6
```

(또는)

```
# shutdown -r now
```

systemctl을 사용해서 재기동이 가능하다.

```
# systemctl reboot
```

## 2.4 인스톨 후 기본 설정 작업

### 2.4.6 런레벨 [ Run Level ]

- 리눅스는 시스템이 기동될 때 런레벨(Run Level: 실행레벨) 값을 참조한다.
- 다음 런레벨은 리눅스 서버 관리자가 알아야 할 값들이다.
  - 0 : 시스템 정지
  - 1 : 싱글 유저 모드로 기동 (시스템 복구시 사용)
  - 2 : 멀티 유저 (네트워크를 사용하지 않는 텍스트 멀티 유저모드)
  - 3 : 멀티 유저 (일반적인 셸 기반의 텍스트 멀티 유저 모드)
  - 4 : 사용하지 않음
  - 5 : GUI 멀티 유저 모드이다.(x-window)
  - 6 : 시스템 재기동
- 일반적으로 서버를 운용할 때는 3 또는 5를 지정한다.
- 보통 서버에서는 GUI가 필요가 없기 때문에 3를 지정하는 것이 보통이다.
- RHEL7에서는 init이 systemd 로 대체되어 숫자 기반의 런레벨이 아니라 각 런레벨에 대한 설정 세트를 통해서 런레벨을 변경 한다.

```
#systemctl rescue (런레벨1)
```

```
#systemctl isolate multi-user.target (런레벨2)
```

```
#systemctl isolate runlevel3.target (런레벨2)
```

```
#systemctl isolate graphical.target (런레벨5)
```

```
#systemctl isolate runlevel5.target (런레벨5)
```

## 2.4 인스톨 후 기본 설정 작업

– /etc/inittab에 설정된 다음과 같은 내용을 리눅스가 기동될 때 참고하게 된다.

```
id:3:initdefault:
```

– 다음 명령으로 내용을 확인해 보자

```
[root@localhost ~]# cat /etc/inittab
# inittab is no longer used when using systemd.
#
# ADDING CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
# Ctrl-Alt-Delete is handled by /usr/lib/systemd/system/ctrl-alt-del.target
#
# systemd uses 'targets' instead of runlevels. By default, there are two main targets:
#
# multi-user.target: analogous to runlevel 3
# graphical.target: analogous to runlevel 5
#
# To view current default target, run:
# systemctl get-default
#
# To set a default target, run:
# systemctl set-default TARGET.target
#
```

## 2.4 인스톨 후 기본 설정 작업

마찬가지로 RHEL7 에서 systemd로 대체되어 이 부분도 변경되었다. /lib/systemd/system 파일의 target 파일을 의 설정 내용을 참고한다.

```
[root@localhost ~]# ls -l /lib/systemd/system | grep runlevel

lrwxrwxrwx. 1 root root 15 2월 11 17:28 runlevel0.target -> poweroff.target
lrwxrwxrwx. 1 root root 13 2월 11 17:28 runlevel1.target -> rescue.target
drwxr-xr-x. 2 root root 49 2월 11 17:28 runlevel1.target.wants
lrwxrwxrwx. 1 root root 17 2월 11 17:28 runlevel2.target -> multi-user.target
drwxr-xr-x. 2 root root 49 2월 11 17:28 runlevel2.target.wants
lrwxrwxrwx. 1 root root 17 2월 11 17:28 runlevel3.target -> multi-user.target
drwxr-xr-x. 2 root root 49 2월 11 17:28 runlevel3.target.wants
lrwxrwxrwx. 1 root root 17 2월 11 17:28 runlevel4.target -> multi-user.target
drwxr-xr-x. 2 root root 49 2월 11 17:28 runlevel4.target.wants
lrwxrwxrwx. 1 root root 16 2월 11 17:28 runlevel5.target -> graphical.target
drwxr-xr-x. 2 root root 49 2월 11 17:28 runlevel5.target.wants
lrwxrwxrwx. 1 root root 13 2월 11 17:28 runlevel6.target -> reboot.target
-rw-r--r--. 1 root root 761 11월 20 13:49 systemd-update-utmp-runlevel.service

[root@localhost ~]#
```

## 2.4 인스톨 후 기본 설정 작업

따라서, 부팅 시 런레벨은 다음 명령으로 설정하고 현재 설정 상태를 알아볼 수 있다.

```
[root@localhost ~]# systemctl get-default  
multi-user.target
```

```
[root@localhost ~]# systemctl set-default graphical.target  
Removed symlink /etc/systemd/system/default.target.  
Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/graphical.target.
```

```
[root@localhost ~]# systemctl set-default multi-user.target  
Removed symlink /etc/systemd/system/default.target.  
Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/multi-user.target.
```

```
[root@localhost ~]# systemctl get-default  
multi-user.target
```

```
[root@localhost ~]#
```

## 2.4 인스톨 후 기본 설정 작업

### 2.4.7 패키지 설치 및 업데이트

- yum 커맨드를 사용해서 패키지 업데이트 및 새로운 패키지 설치가 가능하다.
- 다음 명령으로 yum 레포지토리 구성 내용을 확인해 보자.

```
[root@localhost ~]# yum repolist
```

```
Loaded plugins: fastestmirror
```

```
Loading mirror speeds from cached hostfile
```

```
* base: centos.mirror.cdnetworks.com
```

```
* extras: centos.mirror.cdnetworks.com
```

```
* updates: centos.mirror.cdnetworks.com
```

repo id	repo name	status
base/7/x86_64	CentOS-7 - Base	9,007
extras/7/x86_64	CentOS-7 - Extras	191
updates/7/x86_64	CentOS-7 - Updates	552
repolist: 9,750		

```
[root@localhost ~]#
```

- 네트워크를 통해 인스톨 및 업데이트가 가능하다.
- 인스톨 CD의 패키지가 업데이트 되었을 수 있기 때문에 업데이트하는 것이 좋다.

## 2.4 인스톨 후 기본 설정 작업

### 설치 패키지 업데이트(update)

```
[root@localhost ~]# yum update
```

```
Loaded plugins: fastestmirror
```

```
Loading mirror speeds from cached hostfile
```

```
* base: centos.mirror.cdnetworks.com
```

```
* extras: centos.mirror.cdnetworks.com
```

```
* updates: centos.mirror.cdnetworks.com
```

```
Resolving Dependencies
```

```
--> Running transaction check
```

```
---> Package bind-libs-lite.x86_64 32:9.9.4-29.el7 will be updated
```

```
---> Package bind-libs-lite.x86_64 32:9.9.4-29.el7_2.2 will be an update
```

```
---> Package bind-license.noarch 32:9.9.4-29.el7 will be updated
```

```
...
```

```
...
```

```
...
```

```
...
```

```
...
```

```
Transaction Summary
```

```
=====Install 1 Package
```

```
Upgrade 26 Packages
```

```
Total download size: 68 M
```

```
Is this ok [y/d/N]:
```

확인 메시지에 **y**를 입력하면 업데이트를 진행 한다.

커널 업데이트를 포함 하는 경우 재부팅 한다.



## 2.4 인스톨 후 기본 설정 작업

### 새 패키지 설치(Install)

```
[root@localhost ~]# yum install crontab
[root@localhost ~]# yum install rdate
[root@localhost ~]# yum install gcc
[root@localhost ~]# yum install make
[root@localhost ~]# yum install wget
[root@localhost ~]# yum install gcc-c++
[root@localhost ~]# yum install cmake
[root@localhost ~]# yum install net-tools
[root@localhost ~]# yum install bind-utils
[root@localhost ~]# yum install psmisc
```

### 그 밖에 자주 사용되는 yum 커맨드

yum clean all	패키지 캐시 파일 삭제
yum list	모든 패키지 표시
yum remove <패키지 명>	패키지 삭제
yum update <패키지 명>	패키지 업데이트

## 2.4 인스톨 후 기본 설정 작업

### 2.4.8 서버 시간 동기화

서버를 운영하다 보면 시간 동기화를 하지 않을 경우 시간이 조금씩 어긋나게 된다. 이를 막아주기 위해서 rdate 혹은 다른 방법을 이용해서 서버시간을 국제 시간에 동기화 시켜 주어야 한다.

#### 실행 스크립트 [time\_sync.sh]

```
#!/bin/bash
rdate -s time.bora.net && date && clock -r && clock -w > /dev/null 2>&1
```

#### cron 작업

- 시스템 운영에 필요한 일상적이고 주기적인 작업을 지정된 시간에 반복적으로 수행하기 위한 목적으로 cron이 제공된다.
- cron은 crond라는 데몬으로 작동하며 리눅스를 설치할 때에도 기본적으로 작동하도록 설정되어 있다.
- cron은 시스템 관리를 위해 꼭 필요한 작업이며, 반드시 구동해야 할 프로그램이다.
- cron은 /etc/crontab 이라는 설정 파일과 /etc 밑에 cron.hourly, cron.daily, cron.weekly, cron.monthly로 구성되며, 실행 시간, 실행 권한, 명령어 등을 적는것만으로 간단하게 설정할 수 있다.

**실행 스크립트 time\_sync.sh를 /etc/cron.hourly 밑에 생성한다.**  
**실행하기 위해 파일 권한을 700을 준다.**

## 2.4 인스톨 후 기본 설정 작업

### 2.4.9 사용자 계정 추가

- root계정은 시스템 설정 변경, 패키지 설치 등 필요할 때만 사용하는 계정이고 로그인(특히, 원격 로그인)등에 사용할 계정으로 보안상 위험한 계정이다.
- root 계정은 시스템에서 어떤 일도 할 수 있는 계정이기 때문에 작업 중 사고가 발생할 가능성이 많다.
- 가급적 root 계정은 꼭 필요할 때만 사용하고 바로 일반 계정으로 돌아와 작업을 해야 한다.
- 따라서, 로그인 계정이 필요하며 계정은 사용자 계정 추가 명령으로 추가할 수 있다.

### 사용자 추가

```
[root@localhost ~]# useradd -g wheel -d /home/kickscar kickscar
[root@localhost ~]# passwd kickscar
kickscar 사용자의 비밀번호 변경 중
새 암호:
새 암호 재입력:
passwd: 모든 인증 토큰이 성공적으로 업데이트되었습니다.
[root@localhost ~]#
```

1. useradd 또는 adduser 명령어를 사용한다.
2. -g 뒤에 그룹 명을 준다. -d 뒤에는 계정 홈 디렉터리이다.
3. 로그인을 하기 위해서는 passwd 명령어를 사용해서 비밀번호를 설정해 주어야 한다.

## 2.4 인스톨 후 기본 설정 작업

### 2.4.10 로그인 스크립트

- 로그인을 하게 되면 몇 가지 설정 스크립트가 실행되고 계정의 환경을 설정하게 된다.
- 순서는 /etc/profile -> ~/.bash\_profile -> ~/.bashrc -> /etc/bashrc

### 보안설정: auto logout 설정하기

해당 시간동안 입력이 없으면 자동으로 로그아웃 시키는 스크립트 추가

```
[kickscar@localhost ~]$ su -  
암호:  
[root@localhost ~]# vi /etc/profile
```

다음 내용 추가

```
export TMOUT = 300
```

환경 설정이므로 설정 후 다시 로그인 하거나 source 명령을 사용한다.

### 보안설정: history 포맷 설정하기

/etc/profile 에 다음 내용을 추가 시켜 보자

```
HISTTIMEFORMAT="%Y-%m-%d_%H:%M:%S [CMD]:"
```

## 2.4 인스톨 후 기본 설정 작업

### 2.4.11 디스크 상태 확인과 mount 정보 보기

#### 디스크 상태 확인

```
192.168.0.8:22 - root@localhost:~ - Xshell 5 (Free for Home/School)
1 192.168.0.8:22
[root@localhost ~]# clear
[root@localhost ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        22G   1.2G   21G   6% /
devtmpfs         992M    0   992M   0% /dev
tmpfs            1001M    0   1001M   0% /dev/shm
tmpfs            1001M   8.4M   993M   1% /run
tmpfs            1001M    0   1001M   0% /sys/fs/cgroup
/dev/sda1        497M  125M   372M  26% /boot
tmpfs            201M    0   201M   0% /run/user/0
tmpfs            201M    0   201M   0% /run/user/1000
[root@localhost ~]#
```

#### 파티션 확인

```
192.168.0.8:22 - root@localhost:~ - Xshell 5 (Free for Home/School)
1 192.168.0.8:22
[root@localhost ~]# fdisk -l

Disk /dev/sda: 25.8 GB, 25769803776 bytes, 50331648 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x0000ec73

   Device Boot      Start         End      Blocks    Id  System
/dev/sda1 *          2048       1026047       512000    83  Linux
/dev/sda2            1026048     5222399       2098176    82  Linux swap / Solaris
/dev/sda3            5222400     50331647      22554624    83  Linux
[root@localhost ~]#
```

## 2.4 인스톨 후 기본 설정 작업

### 마운트(mount) 란?

- 특정 디바이스(device)를 사용하기 위해 하드웨어와 디렉토리를 연결하는 작업
- 하드웨어 중 하드디스크의 장치명이 /dev/sda1라고 한다면 이 장치명을 지정된 마운트 포인트 (디렉터리 위치)와 연결하는 작업이 마운트(mount) 이다.

### /etc/fstab

mount 설정을 영구적으로 할 수 있도록 하는 설정파일이다.

```
$ cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Thu Feb 11 17:26:56 2016
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=81980325-9871-4c01-b67f-ff3489670ac3 /          xfs     defaults    0 0
UUID=86bba47e-dca1-466c-9512-433e61220534 /boot      xfs     defaults    0 0
UUID=a0c235d4-c8f8-4dc1-842b-1fa813f8a5e0 swap       swap    defaults    0 0
```

## 2.4 인스톨 후 기본 설정 작업

### blkid

uuid를 파티션으로 바꿔주는 명령이다.

```
[kickscar@localhost ~]$ blkid
```

```
/dev/sda1: UUID="86bba47e-dca1-466c-9512-433e61220534" TYPE="xfs"  
/dev/sda2: UUID="a0c235d4-c8f8-4dc1-842b-1fa813f8a5e0" TYPE="swap"  
/dev/sda3: UUID="81980325-9871-4c01-b67f-ff3489670ac3" TYPE="xfs"
```

### /etc/fstab + blkid

```
/dev/sda3  /      xfs  defaults  0 0  
/dev/sda1  /boot xfs  defaults  0 0  
/dev/sda2  swap  swap defaults  0 0
```

1. 파일 시스템 장치명 /dev/sda3 는 마운트 포인트 (/ 디렉터리)에 연결되어 있다.
2. 파일 시스템 장치명 /dev/sda3 는 xfs 파일 시스템이다.
3. 파일 시스템 장치명 /dev/sda3 는 default(rw, nouser, auto, exec, suid)속성을 가지는 파일 시스템
4. 덤프(백업)가 불가능하며 fsck가 무결성 체크를 하지 않는다.

## 2.4 인스톨 후 기본 설정 작업

### mount 명령

- 현재 시스템에서 마운트 되어 있는 정보를 출력한다.( 이 정보는 /etc/mtab 에 저장)
- 사용자 또는 시스템이 빈번히 마운트(mount)와 마운트 해제(umount)를 하는 정보와 영구적 마운트 정보를 함께 표시한다.

### CD-ROM 마운트 하기

```
[root@localhost cdrom]# mount -t iso9660 /dev/cdrom /mnt/cdrom
```

### CD-ROM 마운트 해제 하기

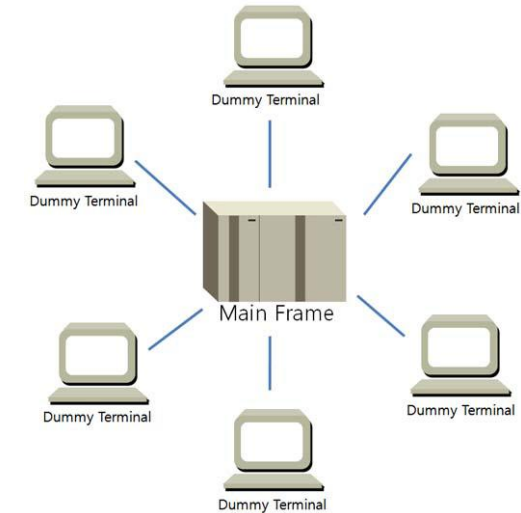
```
[root@localhost cdrom]# umount /dev/cdrom
```



## 2.5 터미널

### 2.5.1 Dummy Terminal

- 70년대 메인프레임 시대에 여러 사용자와 메인 프레임 사이의 인터페이스 역할을 했었다.
- 키보드와 모니터로만 구성되어 있었고 자체 처리 능력은 없었다.
- 명령 프롬프트 같은 텍스트 기반의 터미널이라는 의미에서 콘솔(Console)이라 칭하기도 한다.



### 2.5.2 Remote Terminal(원격 터미널)

- 더미 터미널이 데스크톱 또는 어떤 형태의 컴퓨터로 변화
- 네트워크의 발달로 더미 터미널과 메인 프레임(서버) 사이의 단순한 키보드/모니터 연결이 아닌 네트워크 연결로 클라이언트/서버(C/S) 연결이 가능
- 즉, 네트워크 연결을 통해 한 쪽이 서버, 한 쪽이 클라이언트가 된다.
- 네트워크로 연결된 상태에서 클라이언트가 서버를 제어할 수 있는데 이를 원격 터미널이라 부른다.
- 터미널 연결이 되면, 연결된 서버에 명령을 내리고 프로그램을 실행 시키기 위해 클라이언트는 서버에서 연결 시 만들어 준 셸을 사용하게 된다.
- 원격 터미널 연결을 하기 위해서는 서버와 클라이언트에 네트워크 프로그램이 실행되어야 하고 그리고 두 프로그램 간에 통신을 위한 프로토콜이 필요하다.(telnet, ssh)

## 2.5 터미널

### 2.5.3 터미널[ CLI, TUI] 의 장점

- 간편하고 빠르다.
- 가볍다. (GUI는 무겁다)
- 강력하다.
- 언제나 사용할 수 있다.(복구 환경, GUI가 지원 안 되는 환경)

### 2.5.4 터미널의 종류

#### 콘솔(console)

- 키보드와 키보드가 연결된 모니터
- 터미널을 부르는 다른 이름으로 터미널과 혼용해서 부르기도 한다.
- 네트워크를 통한 원격 액세스가 일종의 규범처럼 여겨지기 때문에 유용성은 많지 않다.
- 네트워크 나 GUI를 사용하지 않는 환경에서는 필요하다.
- 컴퓨터 로컬 접속은 보안의 이유로도 중요하다.

#### GUI 가상 터미널

- X window에서 가상 터미널을 제공하는 프로그램
- 컴퓨터 바로 앞에 키보드와 모니터를 연결해 놓은 것과 같은 터미널 프로그램
- 다 수의 터미널을 제공하는 것과 같다

## 2.5 터미널

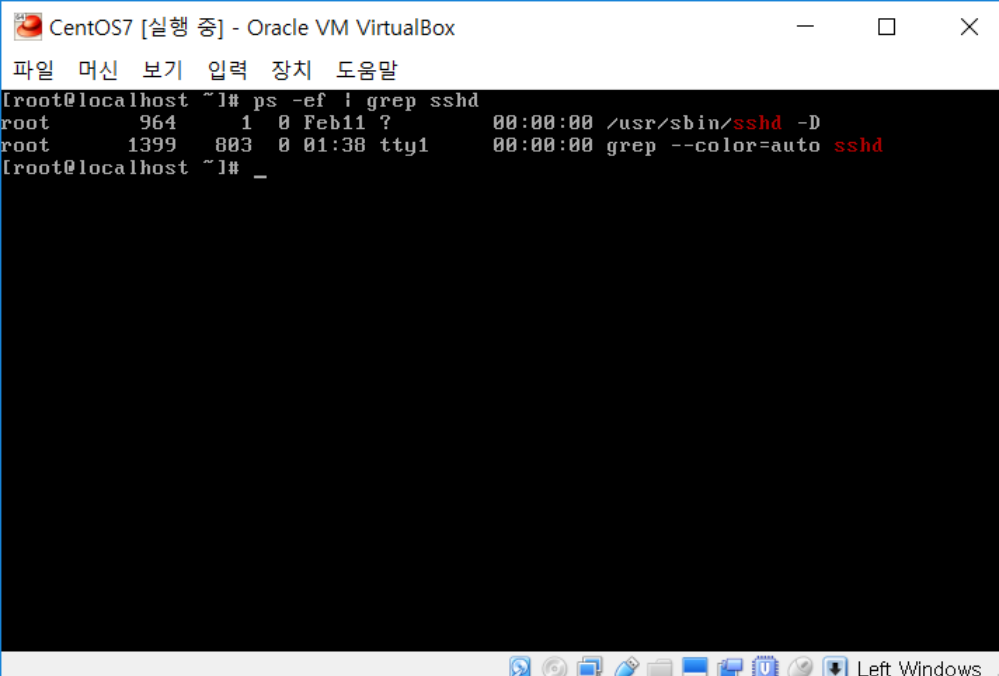
### 원격 터미널(telnet 또는 ssh)

- 가상 터미널이라는 장점을 고스란히 제공하면서 네트워크상의 어떤 컴퓨터에서도 접속할 수 있다.
- 보안상의 이유로 telnet 보다는 ssh를 많이 사용한다.
- ssh는 유닉스 계열을 OS가 설치된 컴퓨터에 원격으로 액세스하는 실질적 표준이라 할 수 있다.
- 두 부분으로 나누어 지는 데 원격 터미널을 요청하고 사용하는 클라이언트와 이를 허용하는 서버 프로그램이다. 무료이고 설정하기도 간편한 편이다.

### SSH 서버 설정하기

#### 2. SSH 서버 실행 여부 확인 하기

```
$ ps -ef | grep sshd
```



The screenshot shows a terminal window titled "CentOS7 [실행 중] - Oracle VM VirtualBox". The terminal displays the command `ps -ef | grep sshd` and its output:

```
root@localhost ~]# ps -ef | grep sshd
root      964      1   0 Feb11 ?        00:00:00 /usr/sbin/sshd -D
root     1399    803   0 01:38 tty1    00:00:00 grep --color=auto sshd
root@localhost ~]# _
```

The terminal window includes a menu bar with options like "파일", "머신", "보기", "입력", "장치", and "도움말". At the bottom, there is a taskbar with various icons and the text "Left Windows".

## 2.5 터미널

### SSH 서버 설정하기

1. SSH 서버가 자동 실행 상태가 아닌지 확인 해 본다.

```
[root@localhost cdrom]# systemctl is-enabled sshd.service
enabled
[root@localhost cdrom]#
```

2. 아니라면 다음 명령어로 자동 실행 상태로 바꾼다.

```
[root@localhost cdrom]# systemctl enable sshd.service
```

3. /etc/ssh/sshd\_config 파일을 열어서 다음 부분을 수정 한다(보안)

root 로 바로 서버접근이 안되도록 설정

PermitRootLogin no

4. sshd 서버를 재시작한다.

```
[root@localhost cdrom]# systemctl restart sshd.service
```

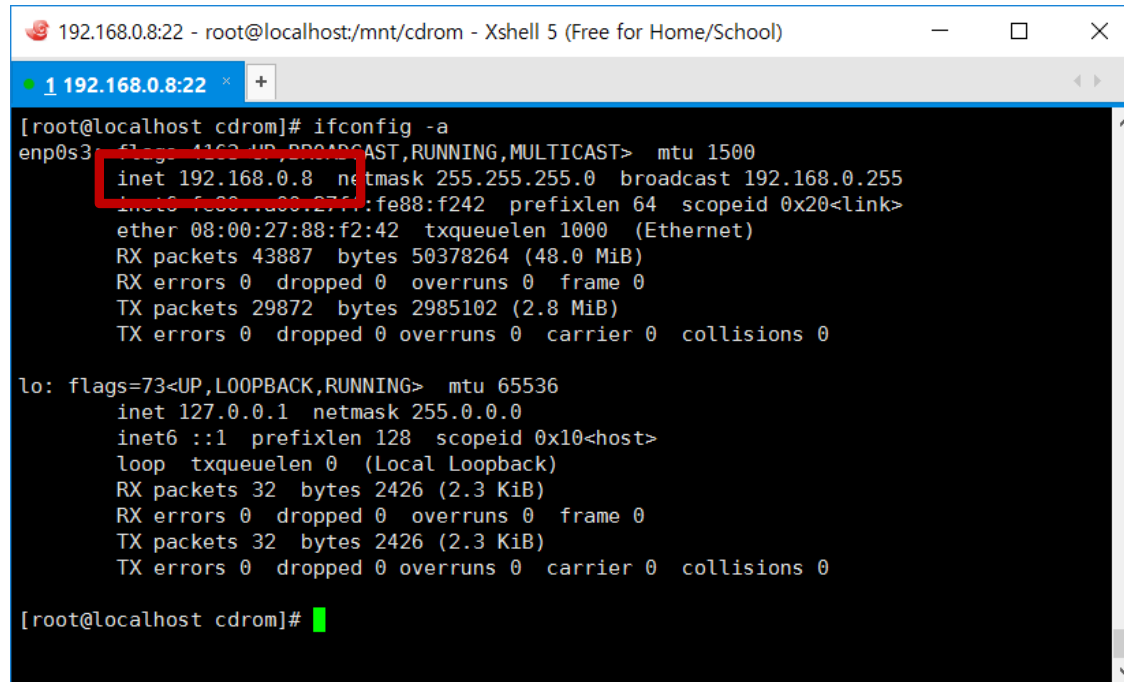
## 2.5 터미널

### SSH 서버 설정하기

#### 3. SSH 서버 IP 확인하기

```
$ ifconfig -a
```

network interface의 상태를 확인한다.



```
[root@localhost cdrom]# ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.8 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::a00:27f:fe88:f242 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:88:f2:42 txqueuelen 1000 (Ethernet)
    RX packets 43887 bytes 50378264 (48.0 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 29872 bytes 2985102 (2.8 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 0 (Local Loopback)
    RX packets 32 bytes 2426 (2.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 32 bytes 2426 (2.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

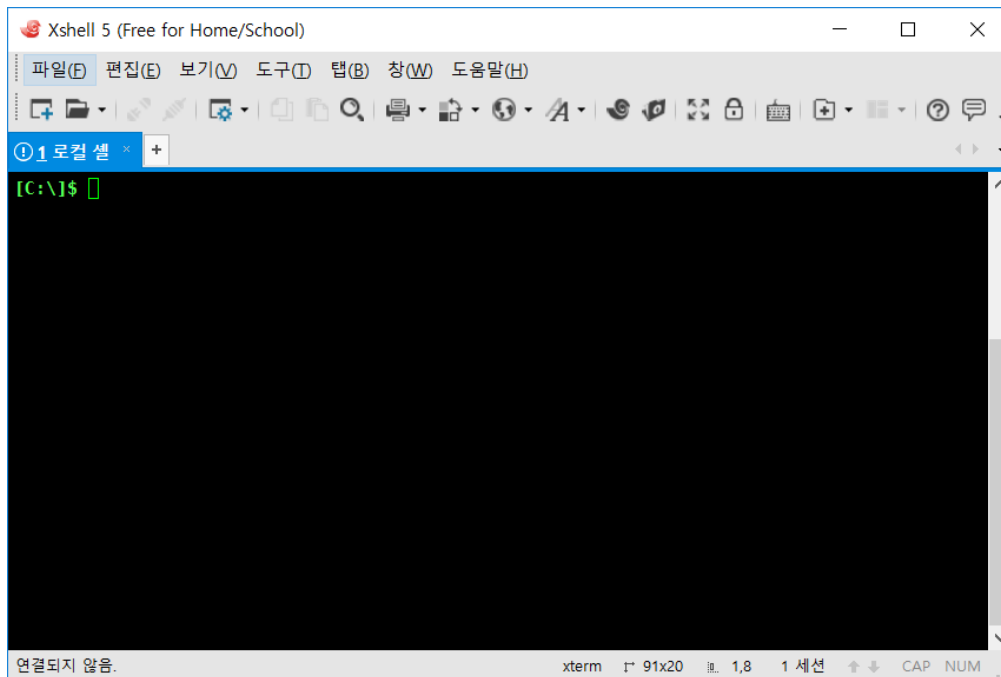
[root@localhost cdrom]#
```

## 2.5 터미널

### SSH 클라이언트 접속하기

#### 2. SSH 클라이언트 프로그램 설치

- 클라이언트 운영체제에 맞는 SSH 클라이언트 프로그램을 구해서 설치한다.
- 거의 모든 운영체제(Unix, Linux, OSX, MS Windows)을 지원하는 SSH 클라이언트 프로그램은 많다.
- MS 윈도우에서는 Putty 또는 Xshell를 많이 사용한다.



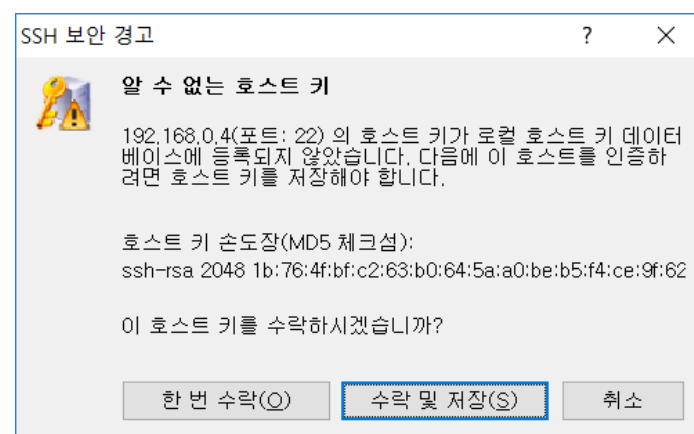
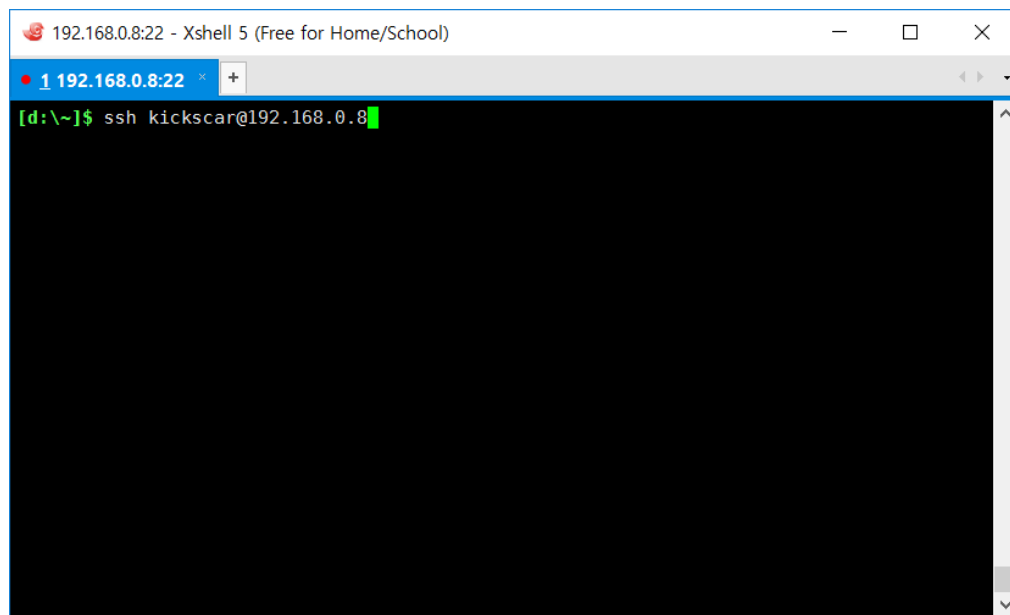
## 2.5 터미널

### SSH 클라이언트 접속하기

#### 2. SSH 서버 접속하기

```
[C:₩]$ ssh myaccount@192.168.0.4
```

접속 계정과 SSH 서버가 실행중인 시스템의 IP를 @ 뒤에 적어주고 엔터를 눌러 연결



# 2.5 터미널

## SSH 클라이언트 접속하기

### 2. SSH 서버 접속하기

SSH 사용자 인증

원격 호스트: 192.168.0.4:22 (%default%)

로그인 이름: pi

서버 종류: SSH2, OpenSSH\_6.7p1 Raspbian-5

아래에서 적절한 사용자 인증 방법을 선택하고 로그인하기 위한 정보를 입력하십시오.

☒ Password(P)

암호(W):

☐ Public Key(U)

사용자 키(K):

찾아보기(B)...

암호

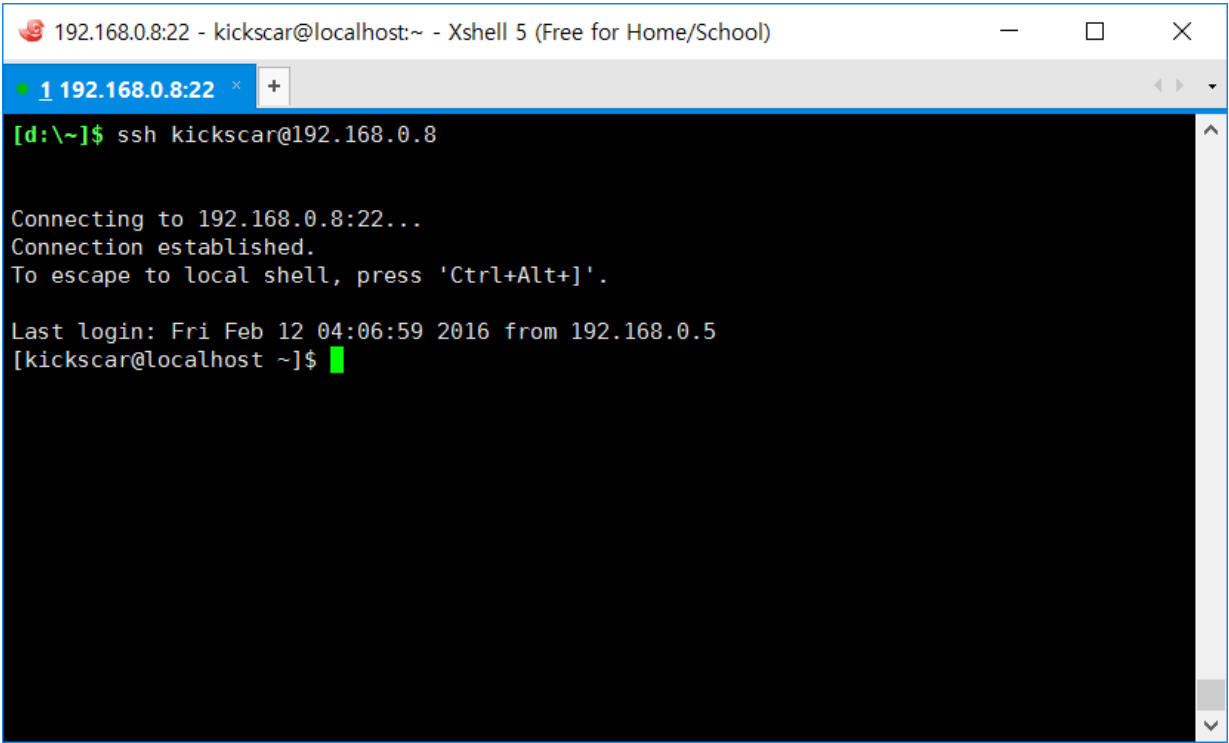
☐ Keyboard Interactive(I)

사용자 인증에 키보드 입력을 사용합니다.

확인

취소

접속 계정의 암호를 입력한다.



접속 성공 화면