



# Report

---

## 오픈소스를 활용한 3-Tier 환경 구축 #1주차

---

- IT 인프라 기초(1)-

---

작성자	코더-정지호, 최예진, 김재현
검수자	송인섭 이사, 윤상훈 수석
작성일	2022-09-18

---

# 목차

---

1. 개요.....	3
2. 리눅스 배포판 사전조사 .....	4
3. 패키지 매니저 종류 및 활용 방법.....	10
4. KERNEL 이란?.....	15
5. 실습 정리 .....	21

# 개요

- 1) 리눅스의 역사, 그리고 배포판 별 특징
- 2) 리눅스 패키지, 패키징 방식, 패키지 관리 방식, 리눅스 패키지 설치 과정
- 3) 커널의 기능, 업데이트 파라미터
- 4) Virtualbox 설치과정 및 cpu/memory/disk 분할 방식과 네트워크 설정에 대해
- 5) CentOS, ubuntu 설치 및 비교
- 6) 커널 업데이트 & 다운그레이드
- 7) CentOS bonding 구성 및 0~6번의 옵션에 대해

## 리눅스 배포판 사전조사

### 1. 리눅스의 탄생배경

리눅스가 등장한 것은 대표적인 서버 운영체제인 유닉스가 개인용 운영체제로 거둔 것이라고 볼 수 있습니다.

#### 1) 유닉스의 탄생배경

1969년 MIT, AT&T 벨 연구소, General Electric에서 켄 톰슨은 멀티태스킹, 멀티 유저를 지원하는 초기 형태의 시분할 운영체제를 만들고자 Multics라는 실험적인 운영체제를 공동으로 개발하는 프로젝트를 통해 진행하여 UNIX를 개발하였습니다.

1971년 처음 출시된 유닉스는 미니 컴퓨터인 PDP-7에서 돌아가도록 만들어졌고 온전히 어셈블리어로 작성되었습니다. 그 당시 하드웨어를 운영하는 프로그램을 그 하드웨어의 어셈블러로 작성하는 것이 일상적이었습니다. 이렇게 UNIX는 기계 의존적이며, 기종간에 호환성이 없는 그런 운영체제였습니다. 그리고 UNIX를 운영체제로 사용하던 PDP-7 또한, 많은 소프트웨어를 제공하지 못하는 상황이었습니다.

1973년, 이러한 상황에서 데니스 리치(Dennis Ritchie)가 C언어를 개발함으로써, 어셈블리 언어로 되어있던 UNIX는 일부 하드웨어 및 입출력 루틴을 제외하고 C언어로 재작성 되었습니다. UNIX의 고급 프로그래밍 언어 구현체의 이용이 가능하게 되었고, UNIX는 이식성과 호환성 있는 시스템으로서 더욱 용이해졌고, 벨 연구소를 중심으로 유닉스 사용자 그룹이 형성되며 빠르게 버전업 되면서 퍼져 나가게 되었습니다. UNIX는 주로 연구와 학습을 목적으로 대학이나, 연구소 등에 무료로 배포되었고, 이를 이용하는 프로그래머들 또한 자연스럽게 서로에게 필요한 프로그램을 공유하는 공동체 분위기가 조성되어 있었습니다. Source 프로그램이 공개되어 있었던 UNIX는 많은 대학들과 연구원들에 의해 연구되어 마침내 상업 시장에 진출하기에 이르렀고, Berkeley Unix(BSD), SYSV와 같은 계열로 분화되고,

SunOS, OSF/1.AIX, HP-UX, Solaris, IRIX, SCOUNIX등과 같은 다양한 버전의 유닉스 운영체제들을 탄생시키는 모체가 되었습니다.

1980년대 초반에 UNIX는 당시 가장 인기 있는 기종인 DEC의 PDP-11와 VAX를 주축으로 발전되고 있었습니다. 한편, MIT의 인공지능 연구소에서는 PDP-10에 탑재하기 위한 시분할 운영체제인 ITS를 개발하였는데, 연구내용을 상업화 하려는 움직임이 있었고, 이로 인해 분열의 조짐이 보이면서, 몇몇 핵심 연구원들은 상업적인 것을 목적으로 하는 회사로 떠나갔습니다. 때마침, DEC이 VAX와 PDP-11을 주력제품으로 삼으면서, PDP-10을 단종시킴으로써 더 이상 ITS의 개발은 이루어질 수 없었습니다.

## 2) GUN프로젝트 시작

1984년 1월, MIT 인공지능 연구소의 연구원으로 ITS 프로젝트에 참여하였던 리처드 스톨먼(Richard Stollman)은 , 소스를 공개하지 못하도록 하는 분위기와 기술을 상업화하려는 조류에 반감을 갖고 온전히 자유로운 소프트웨어로 구성된 "완전한 유닉스 호환 소프트웨어 시스템"을 만드는 목표를 두고 C로 작성된, 그리고 모두에게 공개된 UNIX 시스템을 위해 GUN프로젝트를 시작하였습니다.

1985년에 GNU 프로젝트로 개발된 에디터인 Emacs에 대한 사용자들의 관심이 높아지면서, 스톨먼은 GNU프로젝트 운영을 위해 FSF(Free Software Foundation, 자유 소프트웨어재단)을 설립했습니다. 개발이 진행된 프로그램들은 GNU프로그램들의 배포 라이선스인 GPL하에서 판매되었으며, 판매 수익은 프로젝트를 운영하는데 사용되었습니다. 모든 GNU프로젝트는 FSF를 중심으로 진행되어 갔습니다.

1990년에 GNU 프로젝트는 거의 완성단계에 이르렀으나, 장치 드라이버, 데몬, GUN/허드 이름의 커널 등이 정체되거나 완성되지 못하였습니다. 스톨먼은 Mach를 기반으로 한 커널인 'Hurd'라는 이름의 커널을 개발하기 시작하였으나, 이미 공개된 프로그램들에 대한 지원 등 여러 가지 이유로 GNU 커널의 개발은 좀처럼 진척되지 않았습니다.

### 3) 리눅스 등장

한편 1987년 미닉스는 마이크로 소프트사의 제닉스와 네덜란드 브리제 대학의 전산과 교수, 앤드루 타넨바움에 의해 소형 유닉스 계열의 운영체제로 개발되었습니다. 초기 운영 체제의 원리를 학습하고자 한 학생들과 다른 사용자들을 대상으로 소스코드와 함께 무료로 배포되었고 PC에서 멀티태스킹을 구현하기에 적절하였습니다. 보다 나은 운영체제로 발전시키기 위한 제안을 받았지만 라이선스 조항으로 인해 2000년 4월 라이선스 변경이 될 때까지 자유 소프트웨어로 자리잡지는 못하였습니다.

1991년, 핀란드의 헬싱키에서 당시 교육용으로만 국한되었던 미닉스 라이선스에 실망한 21세 대학생이었던 리누스 토르발스가 GNU시스템에 적합한 커널, 유닉스의 커널을 PC에서 사용가능한 커널로 개발하였습니다. 초기 버전 0.01은 가장 기본적인 커널만을 포함하고 실행조차 되지 않는 아주 미약한 기능으로 발표도 되지 않았으며 최초 공개된 0.02버전은 bash(GNU Bourne Again Shell)와 gcc(GNU C 컴파일러)정도가 실행될 수 있는 수준이었습니다. 이후 개발자들은 GNU 구성 요소들을 리눅스 커널로 통합하는 일을 시작함으로써 온전히 기능하는 자유 운영 체제를 만들 수 있게 되었습니다.

## 2. 각 배포판 별 특징 조사

리눅스 배포판은 리눅스에서 커널과 함께 해당 운영체제 목적에 맞는 여러가지 프로그램을 패키징해서 제공한 것으로. 최초의 배포판 SLS 등장 이후에 수많은 리눅스 배포판이 등장하지만 주로 패키지 관리 기법에 따라 크게 슬랙웨어, 데비안, 레드햇 3종으로 나눌 수 있습니다.

### 1) SLS

최초의 리눅스 배포판인 SLS는 1992년 5월에 피터 맥도날드에 의해 만들어진 소프트 브랜딩 시스템입니다. SLS는 리눅스 커널과 다양한 GNU프로그램, 기본적

인 유틸리티와 X윈도우까지 탑재하여 하나의 운영체제로 탄생하였습니다. 출시 당시에 가장 인기 있는 배포판이었지만 크고 작은 여러 버그들과 응용프로그램의 설치 및 제거, 검증 등의 패키지 관리에 문제가 있었습니다. 이것은 곧 다른 리눅스 배포판의 등장을 알리는 계기가 되었습니다..

## 2) 슬랙웨어 계열

### -슬랙웨어 리눅스

슬랙웨어 리눅스는 현재까지 가장 살아있는 가장 오래된 배포판으로 패트릭 볼커딩이 SLS에 존재하는 버그를 잡기 위해 시작하여 이 결과로 만들어진 것입니다. 소프트웨어를 최장단에서 최대한 수정되어 배포되는 형식으로 사용자들이 내장된 프로그램을 사용하기에는 편리합니다. 하지만 패키지를 새롭게 적용하거나 수정하기에는 힘들다는 단점이 있습니다. 슬랙웨어 계열에 속하는 배포판에는 SuSE, Porteus, Vector Linux, Salix Os 등이 있습니다.

### -SUSE 리눅스

수세 리눅스는 독일에서 출시된 배포판이고, 유럽에서 주로 개발되었습니다. 신기술, 신기능에 대한 적용이 빠르고, 초기부터 IBM Power System과 IBM zSystem 포팅을 가장 적극적으로 지원하여 레드햇 엔터프라이즈 리눅스보다 높은 안정성을 보여줍니다. 시스템 관리나 SAP솔루션과 같은 특정솔루션에 대한 페이지 캐시 메모리 관리 등 연계기능 적용이 빠릅니다. 저장소가 우분트나 민트에 비해 많지 않고, 국내에 정보가 많지 않다는 단점이 있습니다.

## 3) 레드햇 계열

### -레드햇 리눅스

레드햇은 1993년에 창립된 리눅스 배포판 제작회사로 레드햇 리눅스를 제작하였습니다. 초기에는 일반사용자가 사용하기 쉽게 리눅스 커널을 중심으로 GNU프

로젝트, BSD, X11 등의 소프트웨어를 조합하여 공급하는 형태로 리눅스 초기의 확산에 많은 공헌을 해왔습니다. 레드햇 리눅스는 RPM 및 YUM이라는 패키지 관리 도구를 제공합니다. GPL 소프트웨어를 중심으로 구성하여 자체 제작 소프트웨어도 GPL로 공개하고 있기 때문에 레드햇 리눅스를 기반으로 하는 타기업의 배포판도 많이 존재합니다. 현재는 기업용 배포판인 레드햇 엔터프라이즈 리눅스를 기반으로 하는 지원 및 교육 등에서 수익을 올리고 있습니다.

#### -페도라 리눅스

레드햇이 레드햇 9.0 이후 무료로 레드햇을 배포하지 않았는데, 그대신 페도라 프로젝트를 후원함으로써 일반인들이 계속 사용할 수 있게 하였습니다. 무료인 페도라 리눅스는 상용인 RHEL에 포함될 새로운 기술을 미리 시험하는 용도로 사용됩니다. 레드햇에서 개발되는 최신 기술들을 체험할 수 있고, 개발을 위한 도구들이 기본적으로 제공됩니다. 기본적인 잔여가 많다는 단점이 있습니다.

#### -CentOS

레드햇 리눅스와 완전하게 호환되는 무료 기업용 리눅스로, 엔터프라이즈 리눅스와 동일한 소스코드 기반의 커뮤니티에서 새로 빌드해서 제공하지만, RHEL의 소스코드를 포함하지는 않습니다. 레드햇 엔터프라이즈 리눅스에 비해 일부 하드웨어 플랫폼, 드라이버, 특정 기능이 제외됩니다. CentOS의 패키지 관리방법으로 wget주소와 RPM, YUM이 있습니다. 국내 파트너사를 통한 1차 기술지원은 지원받을 수 있지만, 레드햇으로부터 2,3차 기술지원은 받을 수 없습니다.

#### 4) 데비안 계열

##### -데비안 리눅스

데비안은 전부 자유소프트웨어로 구성된 유닉스 계열의 운영체제로 리눅스 커널을 탑재한 데비안 GUN/리눅스, GNU 허드 커널을 탑재한 데비안 GUN/허드,



FreeBSD 커널을 탑재한 데비안 GUN/KFreeBSD, NetBSD 커널을 탑재한 데비안 GUN/ NetBSD 등으로 나뉘며 현재 이 가운데에서 정식판이 존재하는 것은 데비안 GUN/리눅스입니다. 데비안은 dtkg 및 apt라는 독자적인 패키지 관리도구를 사용하여 패키지 설치 및 업그레이드의 단순함을 가집니다. 일단 인스톨을 한 후 패키지 매니저인 APT업데이트 방식을 이용하면 소프트웨어의 설치나 업데이트에서 다른 패키지와의 의존성확인, 보안관련 업데이트 등을 자동으로 설치해줍니다. 데비안은 네트워크 결합 스토리지부터 전화기, 노트북, 데스크탑 및 서버까지 다양한 하드웨어에서 사용할 수 있습니다. 또한 안정성과 보안에 중점을 두며, 다른 많은 리눅스 배포판의 가반으로 쓰이고 있습니다.

### -우분트 리눅스

우분트는 데비안 GNU 리눅스기반으로 만들어졌으며 고유한 데스크탑 환경을 사용하는 리눅스배포판입니다. 전 세계 누구나 어렵지 않게 리눅스를 사용자는 의도로 만들어졌습니다. 우분트는 개인용 데스크탑 환경에 최적화 되어있으며, 자유 소프트웨어에 기반합니다. 그리고 소프트웨어 설치 및 업데이트 방식은 터미널을 사용하는 데비안의 APT 등 뿐만 아니라 그놈(GNOME)의 어플리케이션 매니저 GUI 환경도 사용하고 있습니다. 또한 시스템 관리작업에서 sudo도구를 사용합니다.

## 패키지 매니저 종류 및 활용방법

### 1. 리눅스 패키지

- 리눅스에서는 설치를 도와주는 파일(윈도우의 윈도우 인스톨러처럼)을 통틀어 패키지라고 부른다.
- 다시 말해, 리눅스 패키지란 리눅스 시스템에서 **소프트웨어를 실행하는데 필요한 파일들(실행 파일, 설정 파일, 라이브러리 등)**이 담겨 있는 설치 파일 묶음이다.
- 리눅스 패키지는 소스 패키지와 바이너리 패키지로 나눌 수 있다.
- **소스 패키지**  
 : 소스 코드가 들어 있는 패키지. 컴파일 과정을 통해 바이너리 파일로 만들어야 실행 가능하다.  
 : 설치할 때 컴파일 작업도 진행되므로 설치 시간이 길고 컴파일 작업 과정에서 오류가 발생할 수도 있다.
- **바이너리 패키지**  
 : 성공적으로 컴파일된 바이너리 파일이 들어있는 패키지로, 소스 패키지에 비해 설치 시간도 짧고 오류가 발생할 가능성도 적다. 리눅스의 기본 설치 패키지들은 대부분 바이너리 패키지이다. **Ubuntu**와 **CentOS**의 기본 설치 패키지이기도 하다.  
 -> 그럼에도 소스 패키지를 사용하는 이유는 내가 원하는 대로 소프트웨어를 수정해서 사용할 수 있기 때문이다.  
 : 또한, 바이너리 패키지는 내 컴퓨터 환경과 바이너리 패키지가 컴파일된 환경이 달라서 문제가 발생할 수 있다.

### 2. 리눅스 패키징 방식

- Debian 계열(**Ubuntu** 등)에서 사용하는 패키징 방식: **DEB**  
 (DEB 패키지 파일의 확장자는 \*.deb)
- Red Hat 계열(**CentOS** 등)에서 사용하는 패키징 방식: **RPM**  
 (RPM 패키지 파일의 확장자는 \*.rpm)

### 3. 리눅스 패키지 관리 방식

- 리눅스에서는 패키지를 관리하기 위해 패키지 관리 도구를 사용한다.
- 패키지 관리 도구는 저수준 패키지 도구와 고수준 패키지 도구로 구성된다.

#### 1. 저수준 패키지 도구

: 패키지 파일을 설치하거나 제거할 때 사용하는 도구.

: 각 패키지의 파일을 설치하고 제거할 수는 있어도 **패키지 간의 의존성은 알지 못함.**

: Debian 계열에서는 **dpkg** 사용

: Red Hat 계열에서는 **rpm** 사용

#### 2. 고수준 패키지 도구

: 패키지 파일을 설치하고 제거할 뿐만 아니라 다운로드 가능한 패키지들을 검색하고 **패키지 종속성(Package Dependencies)을 해결하는 도구**(패키지 종속성을 해결한다는 의미는 해당 패키지가 가지고 있는 패키지 간의 종속성을 파악하고 필요하다면 자동으로 설치해준다는 뜻).

: 패키지 종속성이란, 내가 필요한 패키지 이외도 다른 패키지나 라이브러리가 필요한 것을 말한다.

: Debian 계열에서는 **apt-get** 사용

: Red Hat 계열에서는 **yum** 사용

## - Red Hat 기준

### - rpm

: Red Hat 계열의 패키징 방식으로, Windows 계열의 setup.exe 파일과 비슷하게 자동으로 설치를 해주는 프로그램이다.

: 기본 사용 명령어들

설치: **rpm -Uvh [패키지명]**

삭제: **rpm -e [패키지명]**

설치확인: **rpm -qa [패키지명]**

상세정보: **rpm -qi [패키지명]**

### -yum

: RPM 기반의 시스템을 위한 자동 업데이터 및 패키지 설치, 제거 프로그램으로, RPM 과 다르게 인터넷 사용이 필수라 패키지간 종속성 문제를 해결할 수 있다.

: 기본 사용 명령어들

설치: **yum install [패키지명]**

삭제: **yum remove [패키지명]**

업그레이드: **yum update [패키지명]**

목록 확인: **yum list [패키지명]**

yum 데이터베이스 동기화 업데이트: **yum update**

## - Debian 기준

### -dpkg

: 데비안 패키지 관리 시스템의 기초가 되는 소프트웨어다.

: \*.deb 설치파일이 이미 작업환경에서 존재하고 있을 때 아래와 같이 할 수 있다.

설치: **dpkg -i 패키지파일이름.deb**

삭제: **dpkg -r 패키지이름** : 패키지만 삭제 / **dpkg -P 패키지이름** :

설정파일까지 삭제

조회: **dpkg -l 패키지이름** : 패키지를 간략히 조회 / **dpkg -L 패키지이름** :

패키지에 의해 소유된 파일까지 조회 / **dpkg --info 패키지이름.deb** : 패키지 파일에 대한 정보를 보여줌

### -apt-get

: dpkg 와 달리 종속된 프로그램이 만약 작업환경에 미 설치되어 있다면 추가 수동설치 필요없이 자동으로 설치해준다.

: 기본 사용 명령어들

패키지 설치: **apt-get install -f 패키지이름** (f는 --fix-broken 옵션으로, 종속성 관련 문제를 해결할때 사용한다)

새 패키지 목록 업데이트: **apt-get update**

패키지를 설치/업그레이드하여 시스템 업그레이드: **apt-get upgrade**

패키지 삭제: **apt-get remove/purge 패키지이름**

사용하지 않는 패키지 제거: **apt-get autoremove**

내려받은 파일 제거: **apt-get clean (or) apt-get autoclean**

apt-cache는 패키지 검색을 담당한다.

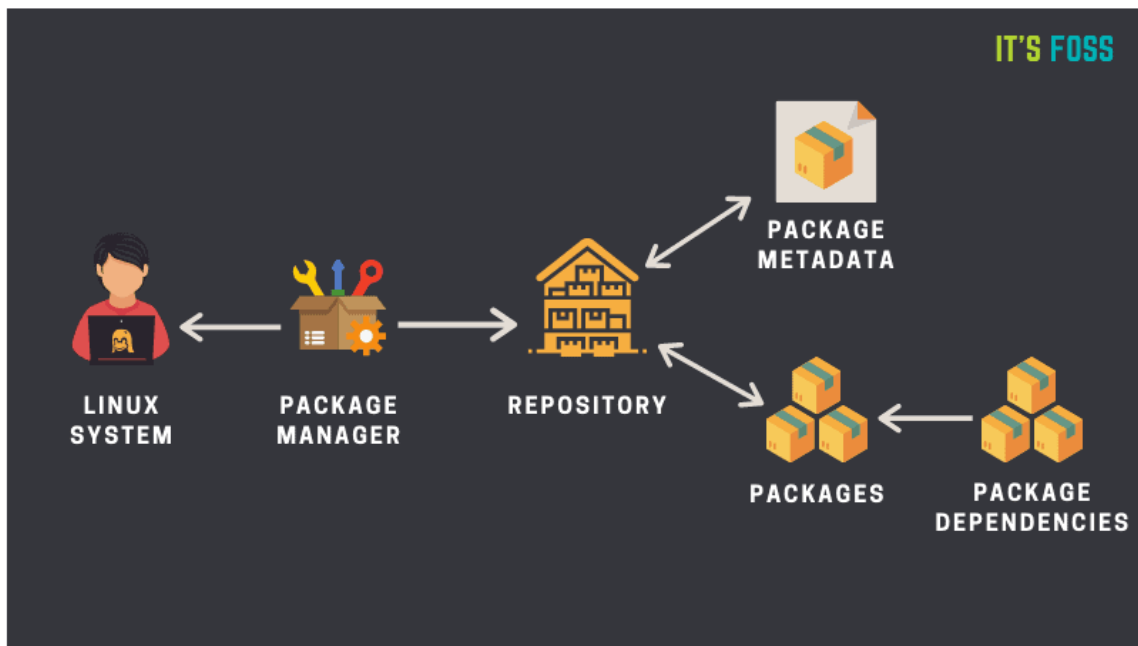
패키지 정보 보기: **apt-cache show 패키지 이름**

패키지 의존성 확인: **apt-cache depends 패키지이름**

패키지의 역의존성 확인: **apt-cache rdepends 패키지이름**

## 4. 리눅스 패키지 설치 과정

- 1) 리눅스 시스템의 패키지 관리자(Package Manager)는 패키지 저장소(Repository)로부터 설치할 수 있는 패키지 정보들을 가진 패키지 메타데이터(Package Metadata)를 로컬 캐시에 저장한다.
- 2) 패키지 관리자(Package Manager)가 특정 패키지의 설치를 명령하면 로컬 캐시를 참조하여 패키지 정보를 찾고 패키지 저장소(Repository)를 통해 패키지를 다운로드한다.
- 3) 이때 패키지는 패키지 종속성(Package Dependencies)이 있을 수 있다. 패키지 관리자(Package Manager)는 종속성을 처리하고 필요한 패키지들을 자동으로 설치한다.



(메타데이터는 구조화된 데이터로, Raw 데이터를 일정한 규칙에 따라 구조화한 정보를 의미한다.)

## kernel이란?

### 리눅스 커널이란

- Linux의 운영 체제(OS)의 주요 구성 요소.  
컴퓨터 하드웨어와 프로세스를 잇는 핵심 인터페이스이며, 두 가지 관리 리소스 사이에서 최대한 효과적으로 통신함.

#### 커널의 기능

- 메모리 관리 : 메모리가 어디에서 무엇을 저장하는데 얼마나 사용되는지 추적.
  - 프로세스 관리 : 어느 프로세스가 중앙처리장치(CPU)를 언제, 얼마나 사용할지 결정.
  - 장치 드라이버 : 하드웨어-프로세스 사이에서 중재자/인터프리터의 역할을 수행.
  - 시스템 호출 및 보안 : 프로세스의 서비스 요청 수신.
- 올바르게 구현된 커널은 사용자가 볼 수 없으며, 커널 공간이라는 자신만의 작업 공간에서 메모리를 할당, 저장되는 모든 항목을 추적한다.
  - 시스템에서 실행되는 코드는 커널 또는 사용자 모드라는 두 모드 중 하나로 CPU에서 실행됨.  
커널 모드 - 하드웨어에 무제한 액세스가 가능.  
사용자모드 - CPU 및 메모리가 SCI를 통해 액세스하는 것을 제한.
  - 위의 작은 세부 사항이 복잡한 작업의 토대가 되며, 이는 프로세스가 사용자 모드에서 실패할 경우 손상이 제한적이며 커널에 의해 복구될 수 있음을 의미한다. 그러나 커널 프로세스의 경우 충돌이 발생하면 시스템 전체가 중지될 수 있다.
  - 안전장치가 마련되어 있고, 경계를 넘기 위해서는 권한이 필요하기 때문에 사용자 프로세스 충돌은 일반적으로 커다란 문제를 유발하지는 않는다.

#### 커널 파라미터

- 리눅스에서 사용되는 시스템 변수.  
커널 변수 값을 제어하여 시스템을 운영 용도에 맞게 최적화할 수 있다.
- 경로는 `/proc/sys` 디렉토리 밑에 존재하며 절대경로의 수정을 통해 설정 변경이 가능하지만, `sysctl`을 통해 간편하게 설정할 수 있다.

- 시스템을 관리하고 제어하며, 메모리와 프로세스에게 값을 할당한다.
- 커널 파라미터의 값을 설정하는 이유는 커널 변수의 값을 제어하여 시스템을 제어할 수 있기 때문.

#### 1. 커널 파라미터 추가

```
# vi /etc/sysctl.conf
kernel.sysrq = 1
```

#### 2. 설정 적용

```
# sysctl -p
```

#### 3. 파라미터 적용 확인

```
# sysctl -a | grep kernel.sysrq
kernel.sysrq = 1
```

or

```
# sysctl kernel.sysrq
kernel.sysrq = 1
```

## Linux 변경점

종류와 버전 확인      `uname -a`

### 레드햇 계열

- CentOS(Community ENTERprise Operating System)
  - Red Hat Enterprise Linux 의 소스코드를 바탕으로 호환을 목표로 개발된 오픈 소스 Linux 배포판.
  - 64 비트 x83 시스템 아키텍처만을 지원함.  
AWS 및 Google 클라우드에 대한 일반적인 초기 이미지도 제공.
- 1) CentOS 6
  - 리눅스 커널 2.6.32 채택.  
커널 모드 디스플레이 드라이버 등이 크게 개선되었다.
  - 가상화 기술로 KVM 이 도입됨.
- 2) CentOS 7
  - System V init 대신 systemd 를 채택. 기존의 init 보다 더 많은 기능을 제공함. 부팅속도가 향상되었고, shell's overhead 또한 감소.
  - 패킷 필터링을 관리하는 방화벽과 Network Manager 도 본격적으로 도입됨.
  - 기본 파일 시스템이 xfs 로 변화됨.



- ext3 에 비해 8 배 많은 inode 를 생성할 수 있다.
  - 메타데이터 저널링 및 빠른 복구 제공.
  - 파일 시스템 포맷이 타 시스템에 비해 압도적으로 빠름.
  - 동일 크기 파일 삭제 액션이 빠르다. 그러나, 대용량 위주의 시스템이기 때문에 작은 단위 파일처리에 대해서는 오히려 속도가 느림.
- 3) CentOS 8
    - `systemctl restart network` 명령으로 변경 사항을 반영할 수 없음.
    - 수명 종료, 및 CentOS Stream 으로 변경됨.
    - 기본 패키지 매니지먼트가 yum 에서 dnf 로 변경됨. 여전히 yum 을 사용할 수 있지만, dnf 의 사용을 권장.
    - Python 3.6, MariaDB 10.3, MySQL 8.0, PostgreSQL 10, PostgreSQL 9.6, Redis 4.0 등을 기본 지원
    - GNOME 의 버전이 GNOME 3.28 로 업그레이드
    - 기본 디스플레이 서버가 Xorg 에서 Wayland 로 변경
    - Cockpit 이라고도 하는 RHEL 8 웹 콘솔을 사용하여 가상 머신을 생성하고 관리할 수 있음
    - QEMU 에뮬레이터에서 샌드박스 기술이 도입되어 가상 머신의 보안이 강화됨.

## 데비안 계열

- Ubuntu
- LTS : Long Term Support, 장기간의 사후지원 기능.  
2 년에 한 번씩 출시되며, 짝수 년도 4 월 버전이 LTS 버전이다. 지원 기간은 5 년이다.
- 1) Ubuntu 12.04 LTS
  - 유니티 5.0 탑재.
    - Lens Quiclists 로 빠른 탐색 가능.
  - CCSM 에 추가항목 탑재
  - Unity Greeter 의 업데이트로 로그인 배경화면 지정 가능
  - 소프트웨어 센터 속도 개선
  - 리듬박스가 다시 기본 음악 플레이어가 되고, 톰보이와 gBrainy 가 기본 설치되지 않음.
  - libreoffice 3.5 베타 2 설치
  - 'Apport' Crash reporting 기본 사용가능.

- 2) 우분투 트러스티(TRUSTY) 14.04 LTS
  - 가장 안정적인 버전.  
기존의 오류가 대부분 픽스되어 배포되었다.
- 3) 우분투 Xenial 16.04 LTS
  - 소프트웨어 센터 변경  
우분투 소프트웨어 센터를 삭제하고 GNOME 소프트웨어 센터를 설치하여 이전에 비해 앱스토어나 플레이스토어 같이 구성이 변경됨.  
설정에 있던 업데이트가 소프트웨어 센터 안으로 옮겨졌으며 전체적으로 시인성이 강화됨.
  - 한/영키를 제대로 지원.
  - AMD 라데온이나 APU 를 사용하고 있는 경우 드라이버를 지원하지 않으므로 직접 설치해야함.
- 4) Ubuntu 18.04
  - Xorg 가 기본 디스플레이 서버가 됨.
  - 리눅스 커널 4.15 지원.
  - Dock 을 제거하는 기능은 없다.
  - Python2 가 사전 설치되어 있다.
  - PHP 는 7.3 버전
- 5) Ubuntu 20.04
  - Xorg 가 기본 디스플레이 서버.
  - 리눅스 커널 5.4 를 지원하며, 기본 ExFAT 지원 및 기타 성능이 향상됨.
  - 32 비트 지원 시스템이 없다.
  - 시각적인 개선.
  - Extensions 응용 프로그램을 사용해 Dock 을 제거할 수 있음.
  - 설치시간과 부팅시간 개선
  - 기본적으로 Python3.
  - PHP 7.4 는 기본 설치 가능 버전입니다.
- 6) Ubuntu 22.04
  - GNOME 42 사용으로 인한 시각적 변화.
  - Grub 부트로더 2.06 버전 사용. 기본적으로 os-prover 기능이 비활성화되어 있다.
  - 새로운 펌웨어 업데이터 앱.
  - 새로운 설치 프로그램
  - Raspberr Pi 에 대한 향상된 지원
  - Wayland 를 기본 디스플레이 서버로 사용.

- 새로운 수평 작업 공간 전환기와 앱을 재정렬할 수 있는 수평 앱 실행기 지원.
- Active Directory 에 대한 많은 지원과 방화벽 백엔드.

## 리눅스

- 리눅스 커널 2.6.32
  - 시스템 성능 개선
    - 선점 형 동작 : 커널 동작 중에도 특정 시간 동안 어떤 프로세스가 디바이스를 점유할 것인지를 결정하는 I/O 스케줄러가 개선되어 한 프로세스가 너무 오래 대기하지 않도록 효율적으로 배분해 시스템에 부하가 많이 걸렸을 때도 사용자 입력에 대해 빠르게 응답함.
    - 하이퍼스레딩 : 하나의 CPU 를 2 개 이상의 CPU 로 활용하는 기능. 부하를 분산하고 최적화.
    - 디바이스 정보
  - 서버 성능 향상
    - NUMA (Non-Uniform Memory Access) 지원  
: 커널 2.4 까지는 여러 개의 CPU 가 메모리를 사용할 겨을 같은 주소에 메모리를 access 하므로 병목현상이 발생함. 2.6 에서는 위상 API 스케줄러를 통해 최대한 효율적으로 가까운 리소스를 활용함으로 병목현상을 없앴으며 비 연속적 메모리 사용이 가능함.
    - 커널에서 디바이스 지원 개수 증가
    - 모듈 서브 시스템의 안정성 향상
  - 데스크톱 성능 향상
    - 주변 장치의 지원과 하드웨어 호환성 향상
- 리눅스 커널 4.15
  - 멜트다운/스펙터 (Meltdown/Spectre)
    - 멜트다운 : '비순차적 명령어 처리'기술의 버그를 악용한 보안 취약점. 보안을 위해 응용프로그램이 CPU 의 캐시 메모리에 접근하지 못했던 기존 하드웨어 구조가 통째로 무너짐.
    - 스펙터 : CPU 속에 있는 수 많은 명령어에서 일어나는 버그를 악용한 보안 취약점. 멜트다운처럼 보안 구조가 통째로 무너지는 것은 아니지만, 처리 중인 데이터 가운데 일부가 해커에게 노출될 수 있음. 추적이 굉장히 어렵다.
    - 문제가 된 비순차적 명령어 처리 기술을 비활성화해 문제해결.

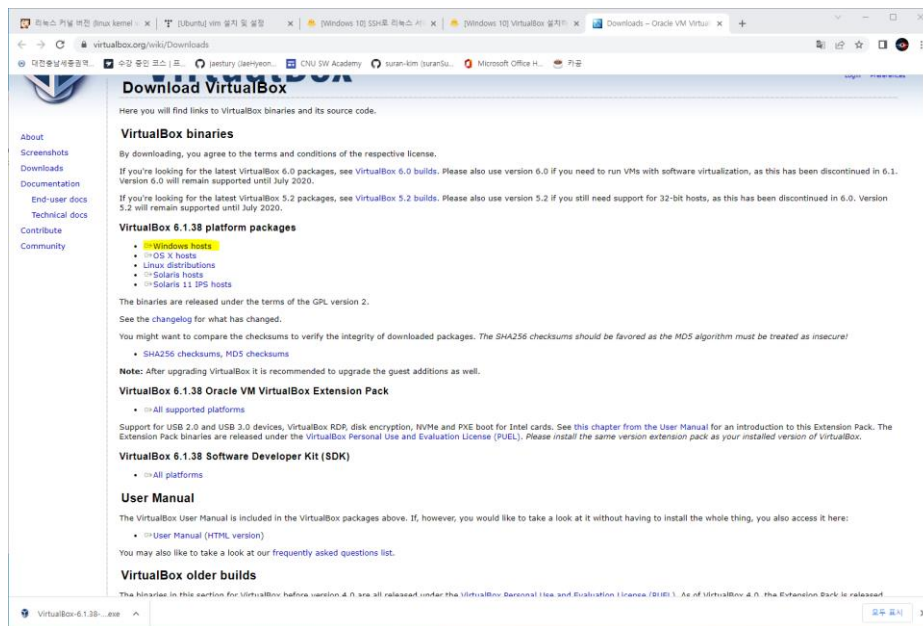
- AMD amdgpu 드라이버 모드 설정 및 더 나은 디스플레이 지원
  - AMD가 최신 그래픽 하드웨어용으로 만든 새로운 드라이버인 amdgpu 드라이버에서 오랫동안 누락되었던 "디스플레이 코드"(132k LoC)가 마침내 포함됨.
- SATA Link Power Management가 있는 시스템의 향상된 전원 관리
- 새로운 아키텍처 : RISC-V
  - RISC-V CPU에 대한 포트의 주요 부분이 포함됨.
- AMD 보안 암호화 가상화 지원
- 사용자 모드 명령 방지 Intel Security 기능 지원
- 리눅스 커널 5.4
  - 커널 lockdown 모드 추가.
    - lockdown 모드 : 유저 프로세스와 커널을 분리, 루트 계정이 코널 코드를 수정하는 것을 방지하는 기능.
  - exFAT 파일 시스템 지원이 이제 커널단에서 이뤄져, exFAT으로 포맷된 드라이브 이용이 편리해짐.
  - AMD Navi 12/14 GPU, AMD Arcturus GPU, AMD Dali APU 지원
  - AMD 뿐만 아니라 인텔 Tigerlake Gen12 그래픽 지원도 추가.
  - 가상화 업데이트
    - Red Hat Enterprise Linux 5.4는 x86\_64 기반 아키텍처에서 KVM (Kernel-based Virtual Machine) 하이퍼바이저를 완전 지원

## 실습 정리

### 1. Virtualbox 설치과정 및 cpu/memory/disk 분할 방식과 네트워크 설정에 대해

#### - Virtualbox 설치하기

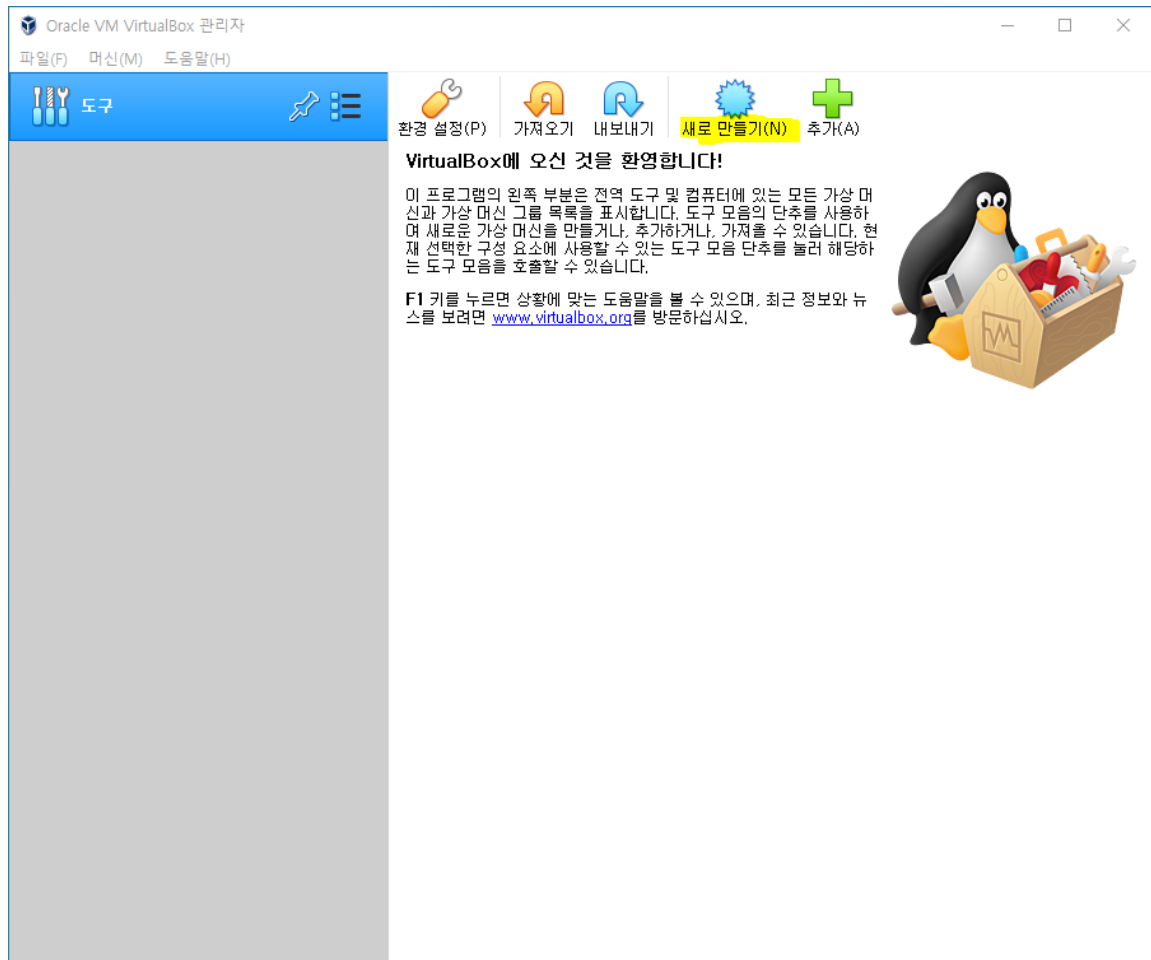
##### 1) Virtualbox 공식 웹사이트에서 바이너리를 다운



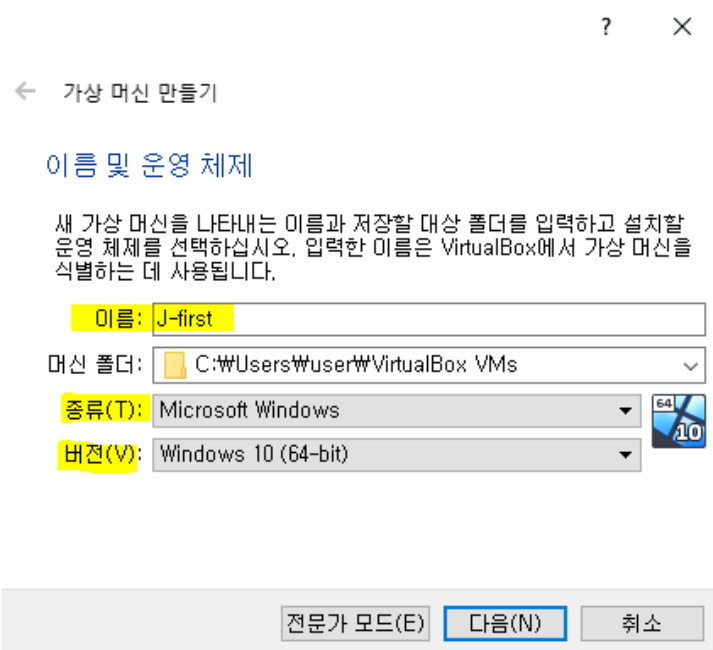
##### 2) 아래의 그림을 통해 설치가 완료됨을 확인

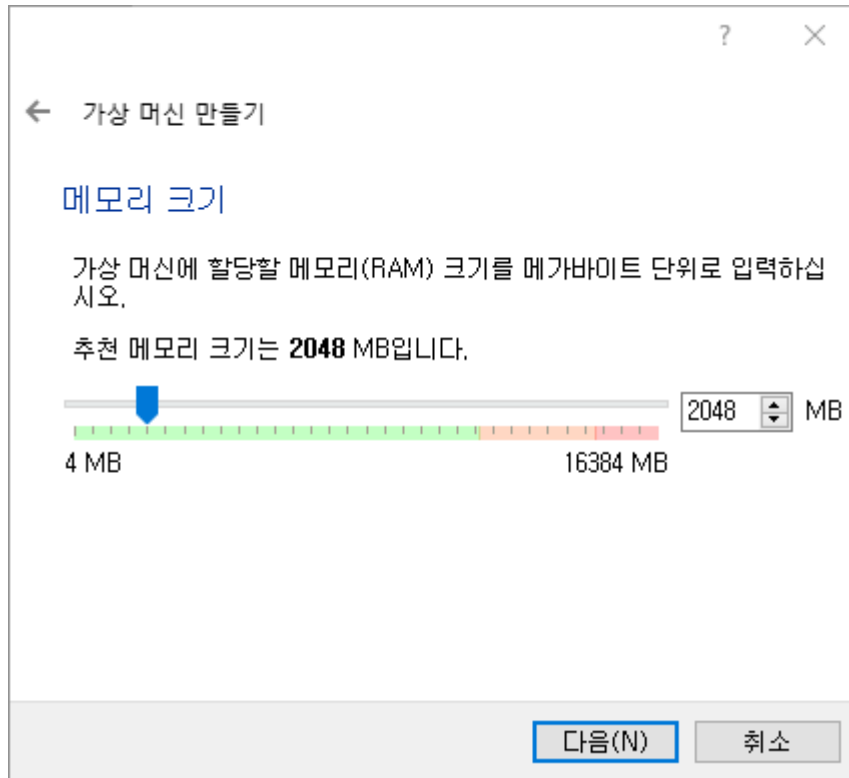


### 3) 새로운 가상머신을 만든다. 새로 만들기를 클릭한다.

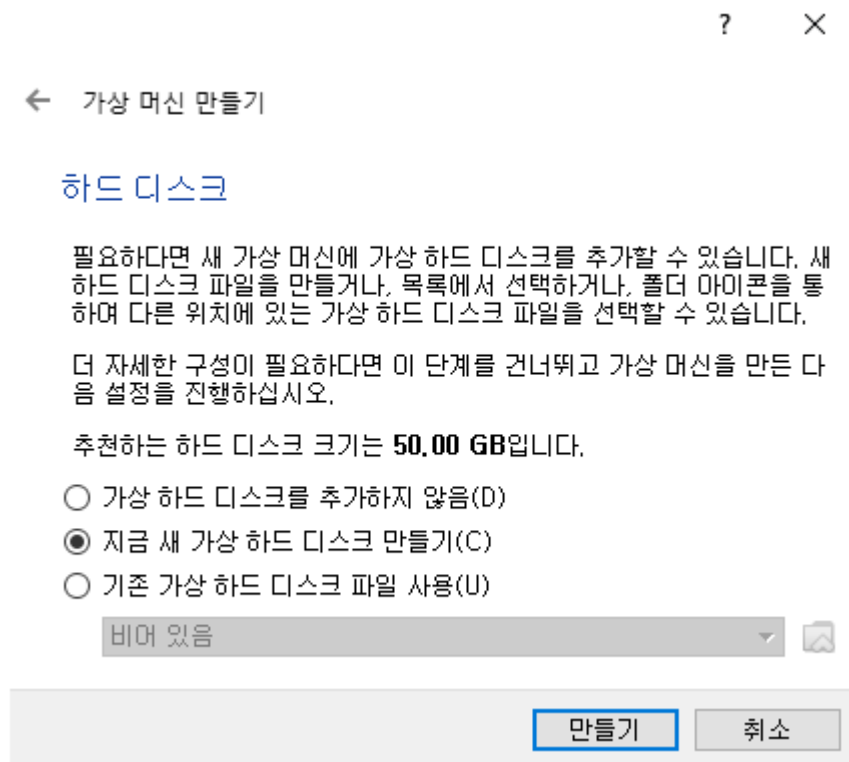


### 4) 이름, 종류, 버전을 설정하고 적당한 메모리를 정해준다.





#### 5) 하드디스크를 만들어준다.(VDI)



? 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

-VDI(Virtual Disk Image): Oracle Virtualbox에서만 사용가능한 포맷방식.

-VHD(Virtual Hard Disk): Oracle Virtualbox, Hyper-V, Xen에서 구동 가능한 포맷 방식

-VMDK(Virtual Machine Disk): Oracle Virtualbox, VMware Player에서 구동 가능한 포맷 방식

? X

← 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 작아지지 않습니다.

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

- ☒ 동적 할당(D)  
☐ 고정 크기(F)

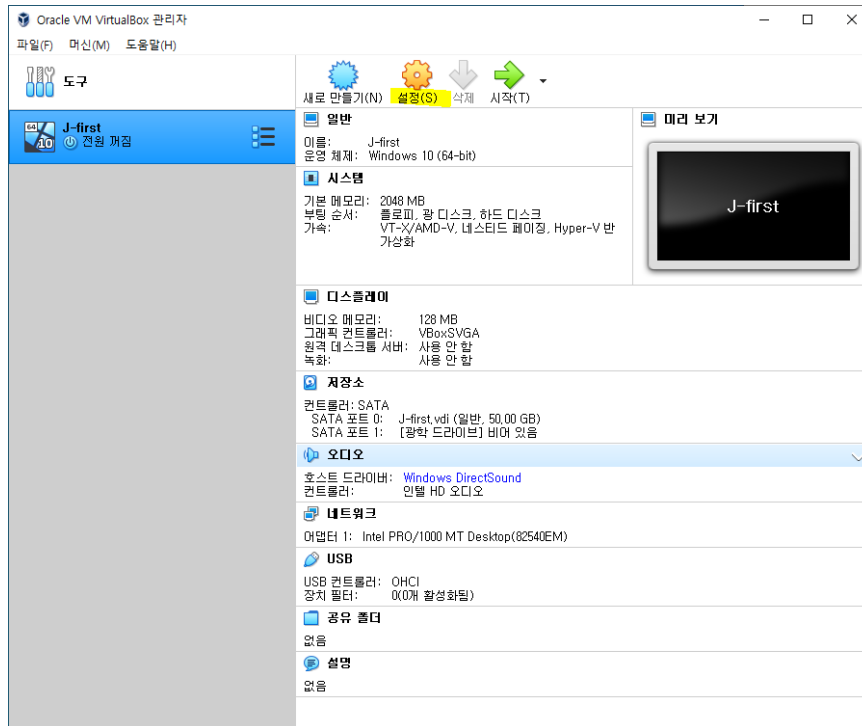
다음(N)

취소

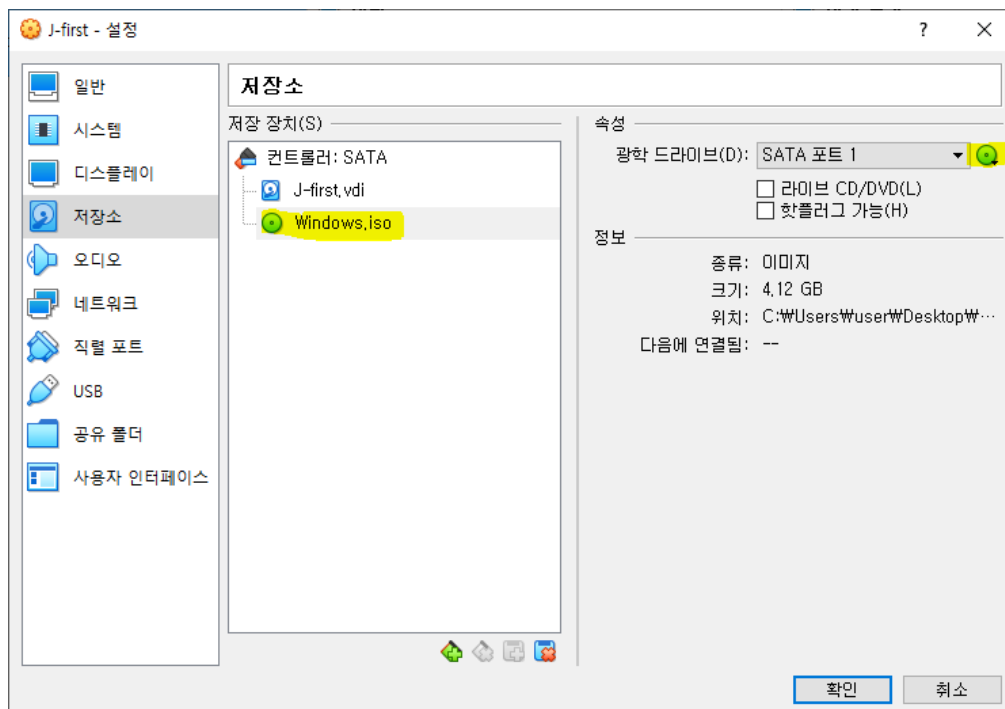


동적할당은 몸 크기가 커질때마다 방을 확장  
고정 크기는 방을 미리 크게 만들어 놓는 것

## 6) 가상머신 완성



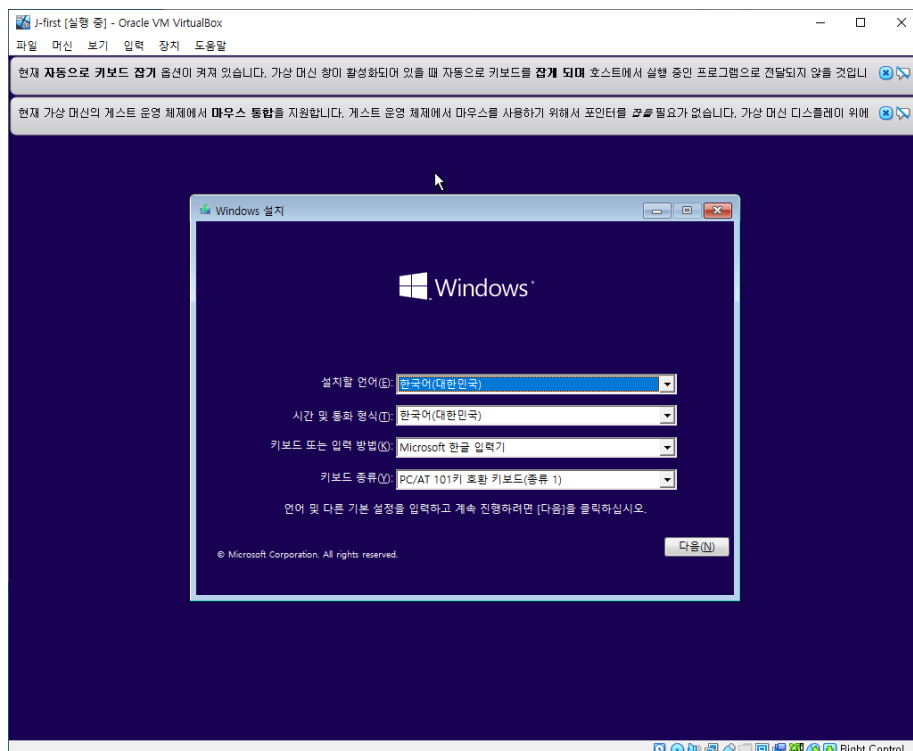
## 7)가상머신에 OS를 띄운다.



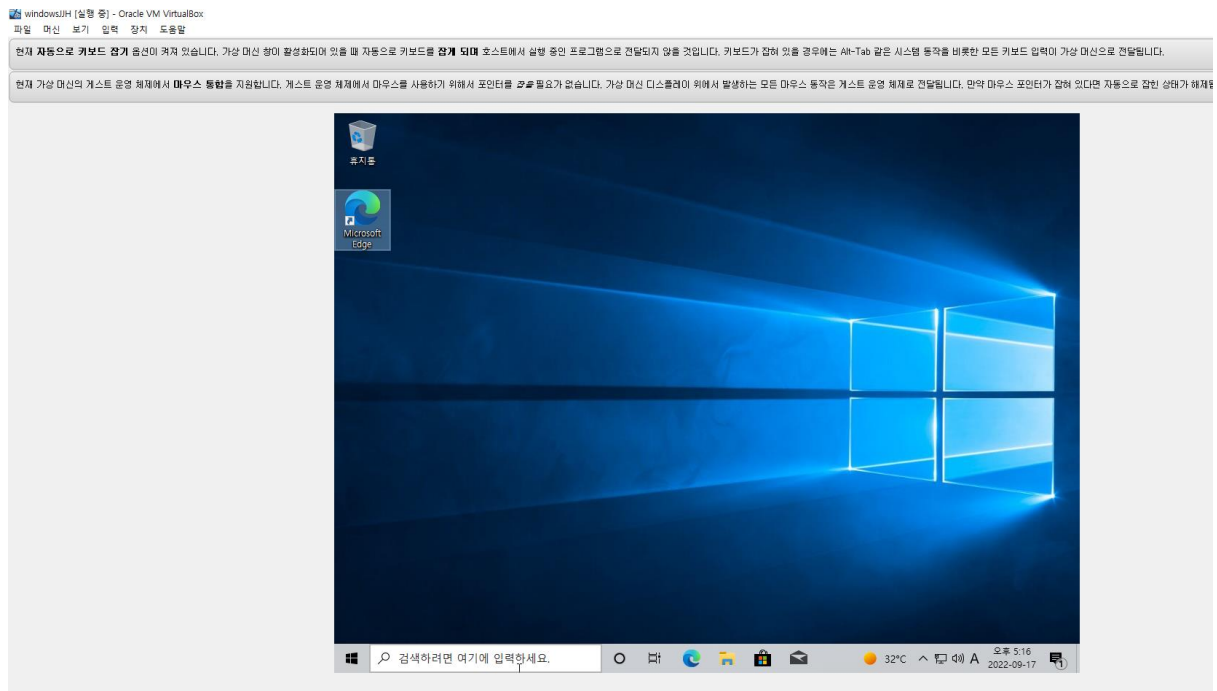
## 8) 실행시켜준다.



## 9) 윈도우 OS 설치 화면이 나온다



## 10) OS 설치 후 화면



## - cpu/memory/disk 분할 방식과 네트워크 설정에 대해

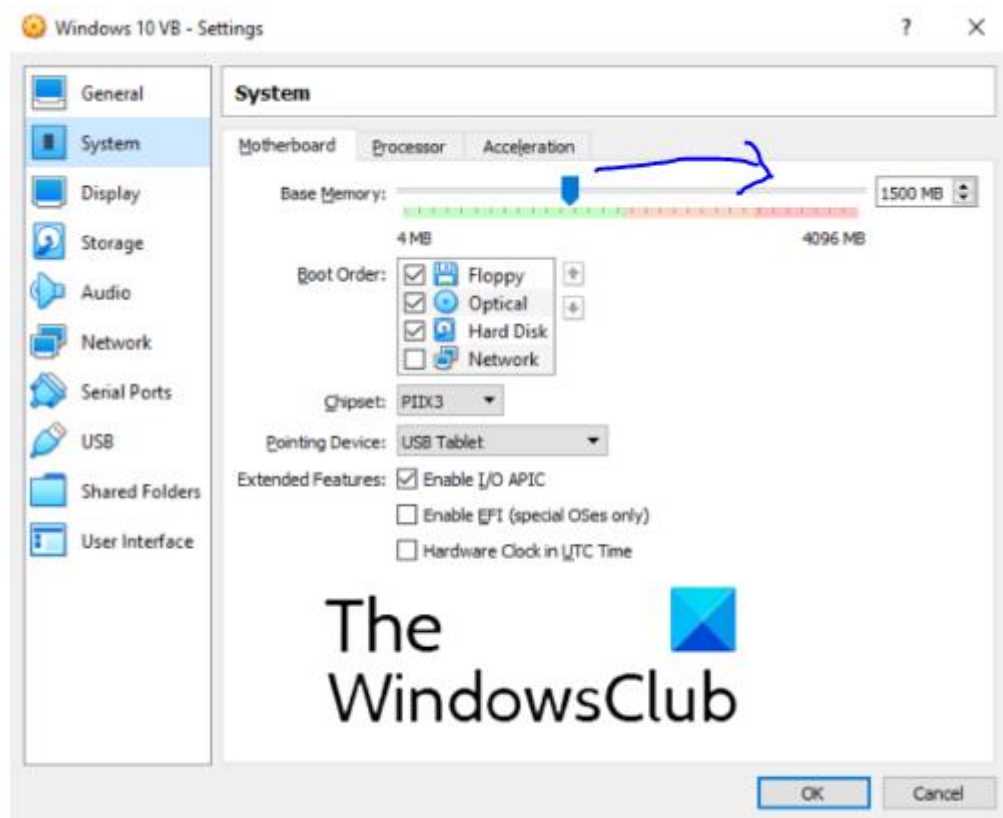
\*가상 머신(VM)을 원활하게 실행하기 위해선 강력한 호스트 시스템이 있어야 하는데 **CPU, 메모리, 디스크 공간**은 호스트 시스템의 성능을 결정하는 세 가지 주요 요소이다.

1. VM에는 많은 처리 능력이 필요하므로, 호스트 시스템에 **다중 코어 중앙 처리 장치(멀티코어 CPU)**가 필요하다.
2. VM의 부드러운 속도를 위해, 시스템에 메모리가 충분히 있어야 한다.  
VM의 **최소 메모리 요구사항은 16GB**인데, 일부 상황에서는 **8GB**가 적합할 수도 있다.
3. VM의 원활한 실행을 위해 **빈 디스크 공간을 충분히 할당**해야 한다.  
그리고 이를 위해서 호스트 시스템의 **디스크에 충분한 공간**이 있어야 한다.

## \*VM의 속도를 높이는 방법

### 1) VM에 더 많은 메모리를 할당

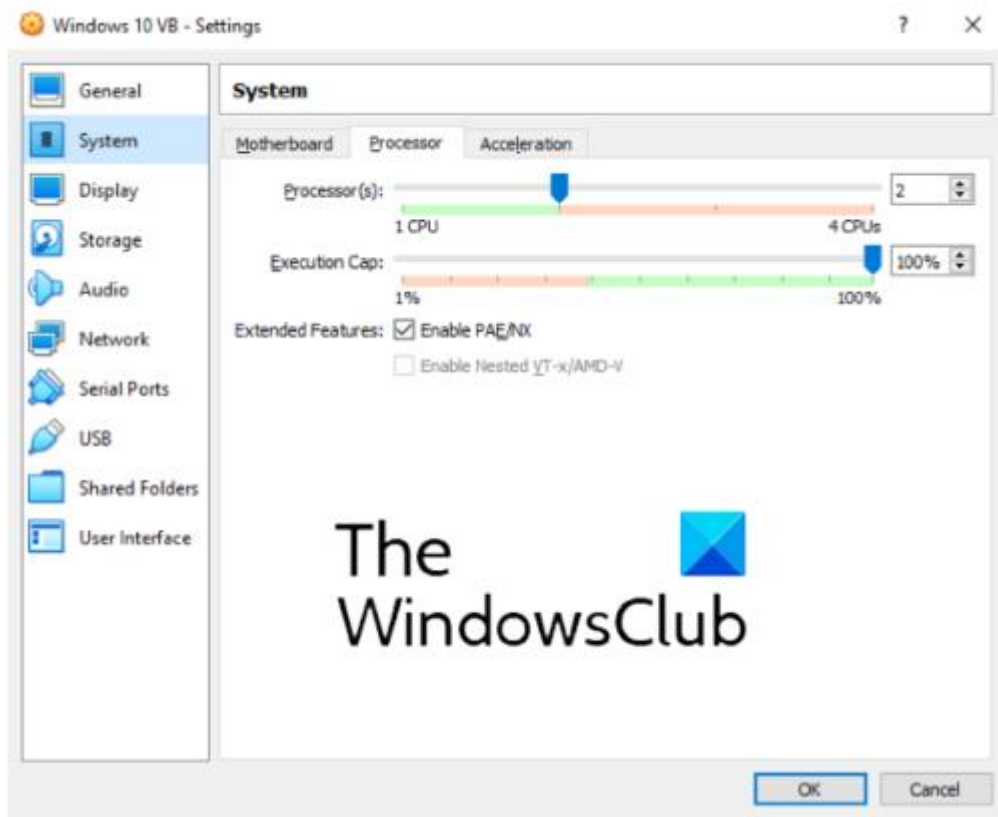
- 물리적 컴퓨터와 마찬가지로 VM의 성능은 할당된 메모리의 양에 따라 달라진다. VM에 더 많은 메모리를 할당할수록 더 부드러운 속도를 얻을 수 있다.



(단 슬라이더를 녹색과 주황색의 교차점에 두면 호스트 시스템에 부정적인 영향을 미칠 수 있으므로, 교차점에 두지 않는다.)

### 2) VM에 더 많은 CPU 할당

- CPU는 VM을 실행하는 데 필요한 모든 계산을 처리한다. 따라서 VM이 더 많은 CPU를 사용하면 더 빠르고 원활하게 실행된다.



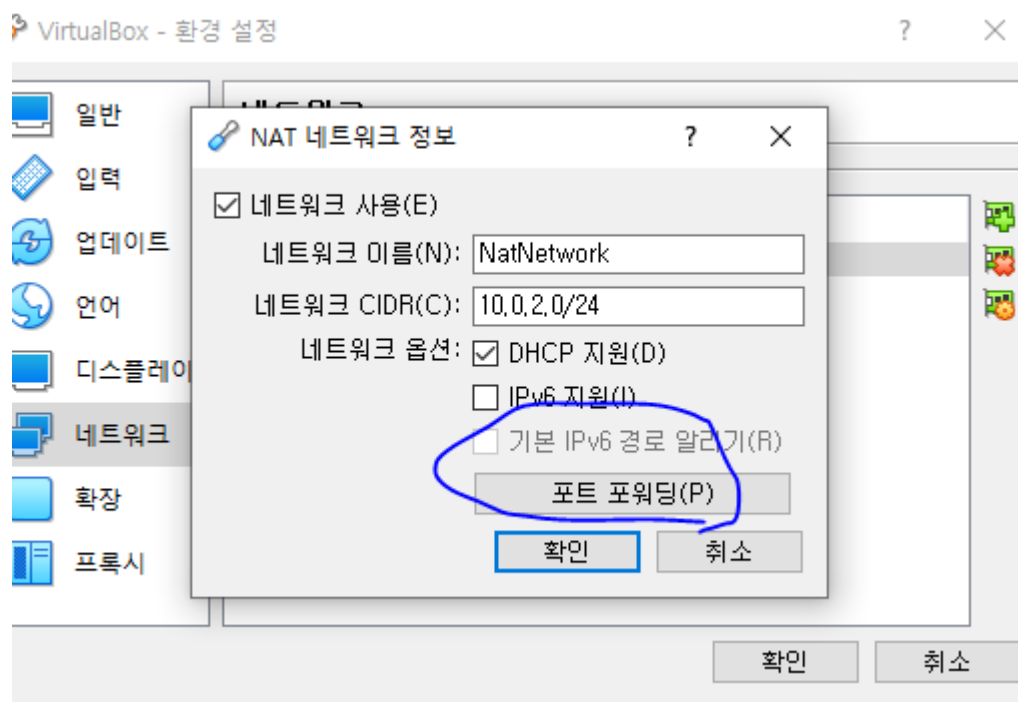
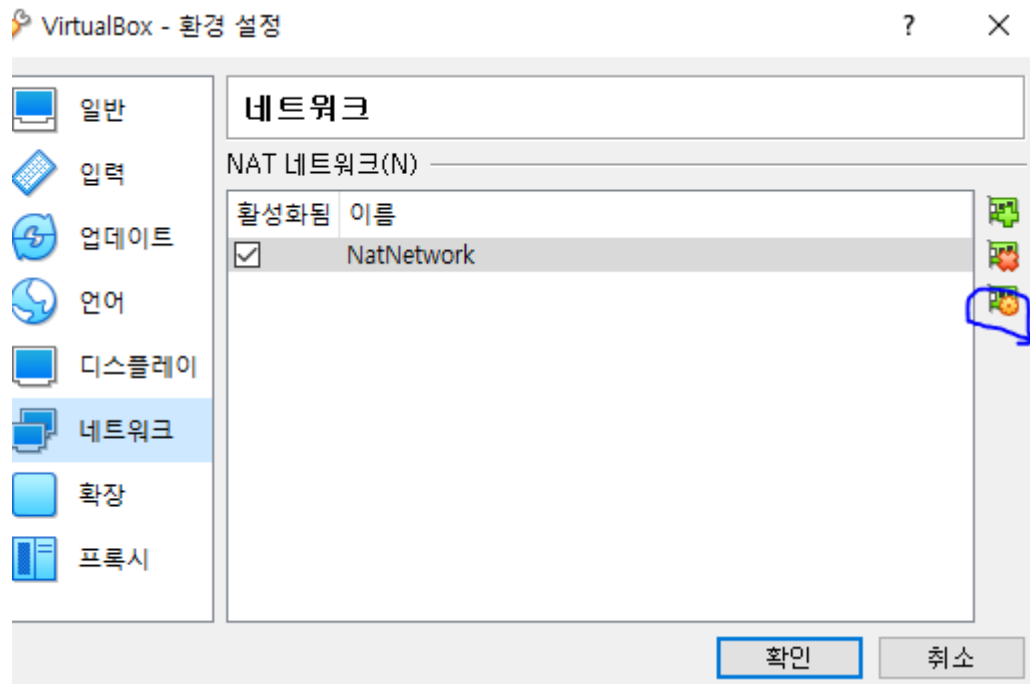
(녹색 영역을 벗어나면 안된다)

- + VirtualBox 를 켤 때만 메모리를 잡아먹고, 켤 때도 쓰는 만큼만 먹는다.
- + VirtualBox 가 Host 의 CPU 를 100% 먹는 경우가 있다.(헛바퀴라고 한다.) 이럴 때 usleep(1) 함수를 이용하여 여유를 주면 된다.

## - 네트워크 설정

### \*포트포워딩

-> VirtualBox 에서 가상 머신의 네트워크를 **NAT Network** 로 하면, **호스트에서 가상 머신으로 접속할 수가 없는데**, 이때 **포트포워딩**을 이용하면 **호스트에서 SSH 접속**을 할 수 있다.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>ipconfig

Windows IP 구성

이더넷 어댑터 이더넷:

    연결별 DNS 접미사. . . . . : 
    링크-로컬 IPv6 주소 . . . . : fe80::d97d:2490:81c2:383c%6
    IPv4 주소 . . . . . : 10.188.191.115
    서브넷 마스크 . . . . . : 255.255.255.0
    기본 게이트웨이 . . . . . : 10.188.191.1

이더넷 어댑터 VirtualBox Host-Only Network:

    연결별 DNS 접미사. . . . . : 
    링크-로컬 IPv6 주소 . . . . : fe80::3960:856a:a74b:c49c%36
    IPv4 주소 . . . . . : 192.168.56.1
    서브넷 마스크 . . . . . : 255.255.255.0
    기본 게이트웨이 . . . . . : 

이더넷 어댑터 vEthernet (Default Switch):

    연결별 DNS 접미사. . . . . : 
    링크-로컬 IPv6 주소 . . . . : fe80::29d4:823f:a656:90f6%26
    IPv4 주소 . . . . . : 172.24.144.1
    서브넷 마스크 . . . . . : 255.255.240.0
    기본 게이트웨이 . . . . . : 

이더넷 어댑터 vEthernet (WSL):

    연결별 DNS 접미사. . . . . : 
    링크-로컬 IPv6 주소 . . . . : fe80::d571:ccfe:63c9:4c9f%32
    IPv4 주소 . . . . . : 172.20.128.1
    서브넷 마스크 . . . . . : 255.255.240.0
    기본 게이트웨이 . . . . . :
```

(호스트 아이디는 192.168.56.1)

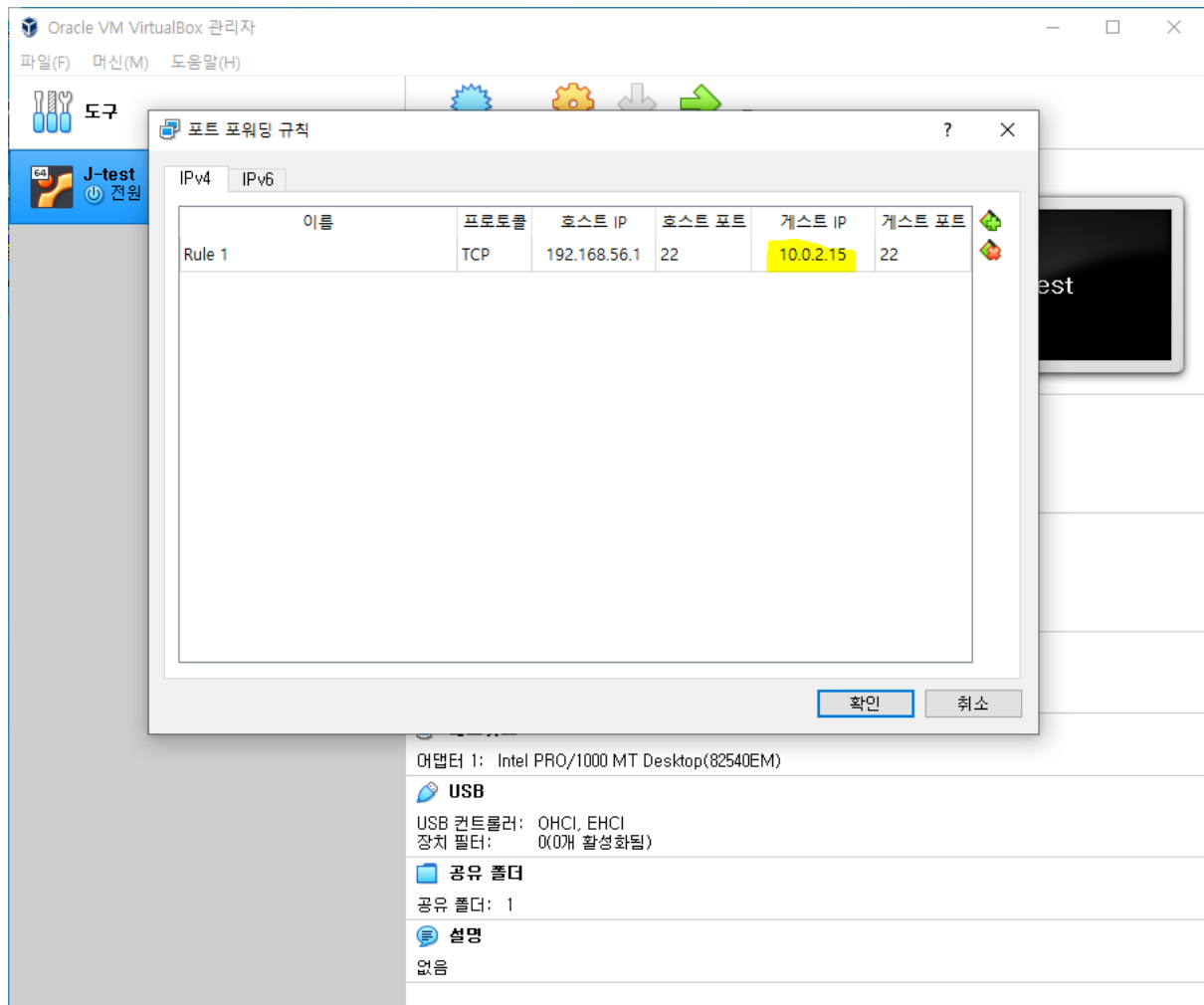
```
J-test [실행 중] - Oracle VM VirtualBox
파일  마신  보기  입력  장치  도움말
Activities  Terminal  9월 17 21 : 05

j-test@jtest-VirtualBox: ~
j-test@jtest-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0  broadcast 10.0.2.255
    inet6 fe80::3448:f596:2698:284  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:a4:df:41  txqueuelen 1000  (Ethernet)
    RX packets 38  bytes 9485 (9.4 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 104  bytes 12391 (12.3 KB)
    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 1000  (Local Loopback)
    RX packets 137  bytes 13762 (13.7 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 137  bytes 13762 (13.7 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

j-test@jtest-VirtualBox:~$
```

(우분투 터미널 활용. 게스트 아이피는 10.0.2.15)



알아낸 정보를 토대로 위와 같이 작성한다(포트는 22로 설정)



## 2. CentOS, ubuntu 설치 및 비교

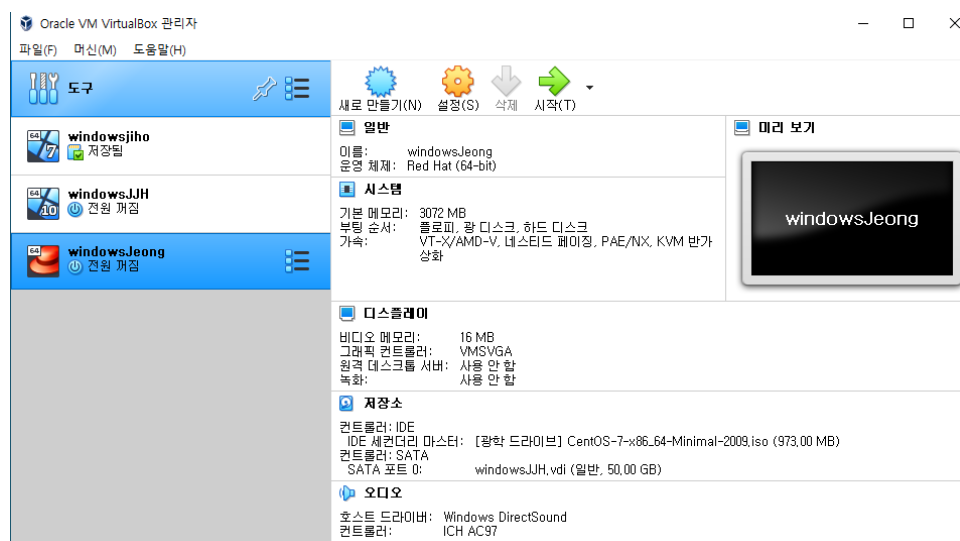
### - CentOS 설치

1) Mimal 버전 iso 파일을 다운.

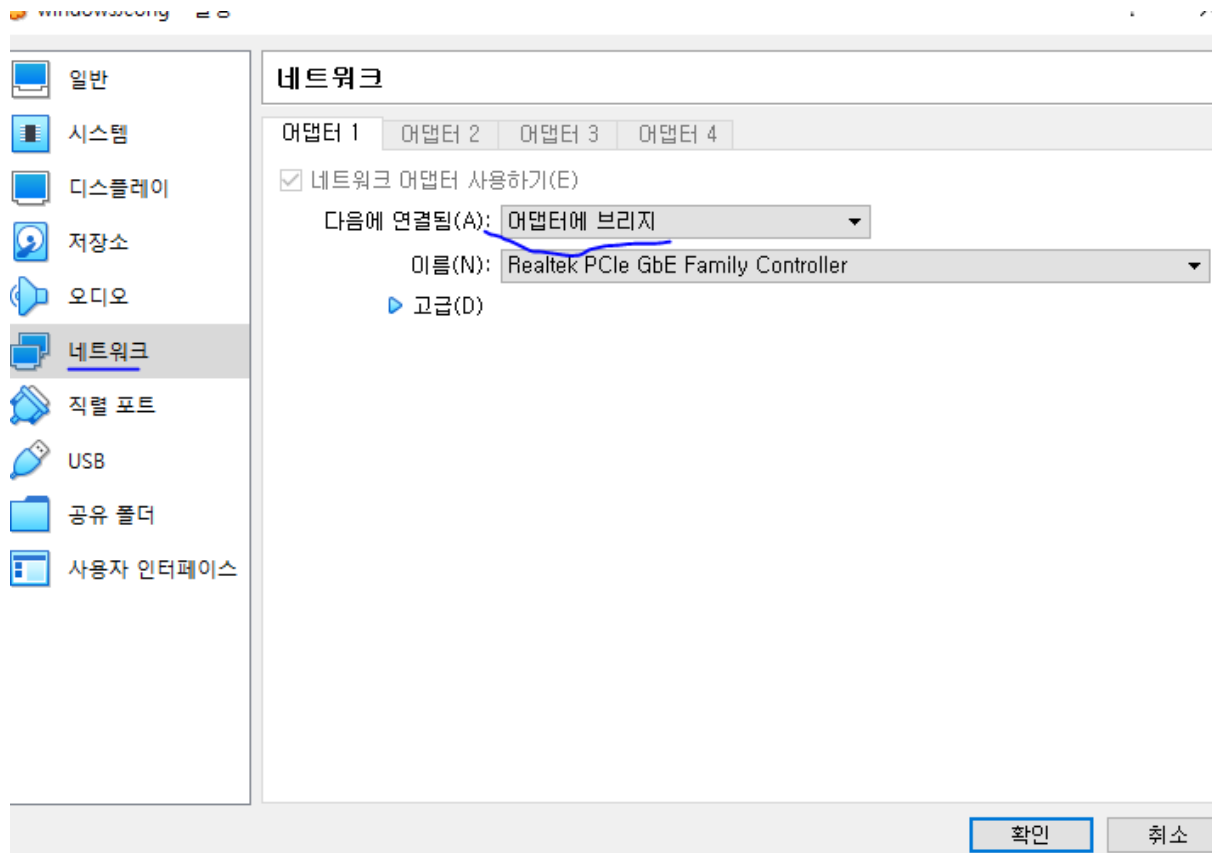
## Index of /centos/7.9.2009/isos/x86\_64

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	
<a href="#">O_README.txt</a>	05-Aug-2022 03:03	2.7K	
<a href="#">CentOS-7-x86_64-DVD-2009.iso</a>	04-Nov-2020 20:37	4.4G	
<a href="#">CentOS-7-x86_64-DVD-2009.torrent</a>	06-Nov-2020 23:44	176K	
<a href="#">CentOS-7-x86_64-DVD-2207-02.iso</a>	27-Jul-2022 00:10	4.4G	
<a href="#">CentOS-7-x86_64-Everything-2009.iso</a>	03-Nov-2020 00:18	9.5G	
<a href="#">CentOS-7-x86_64-Everything-2009.torrent</a>	06-Nov-2020 23:44	381K	
<a href="#">CentOS-7-x86_64-Everything-2207-02.iso</a>	27-Jul-2022 03:09	9.6G	
<a href="#">CentOS-7-x86_64-Minimal-2009.iso</a>	03-Nov-2020 23:55	1.0G	
<a href="#">CentOS-7-x86_64-Minimal-2009.torrent</a>	06-Nov-2020 23:44	39K	
<a href="#">CentOS-7-x86_64-Minimal-2207-02.iso</a>	27-Jul-2022 00:10	1.0G	
<a href="#">CentOS-7-x86_64-NetInstall-2009.iso</a>	27-Oct-2020 01:26	575M	
<a href="#">CentOS-7-x86_64-NetInstall-2009.torrent</a>	06-Nov-2020 23:44	23K	
<a href="#">sha256sum.txt</a>	05-Aug-2022 02:56	703	
<a href="#">sha256sum.txt.asc</a>	05-Aug-2022 02:58	1.5K	

2) iso 적용 + RedHat 계열 선택 그 후 실행(시작)

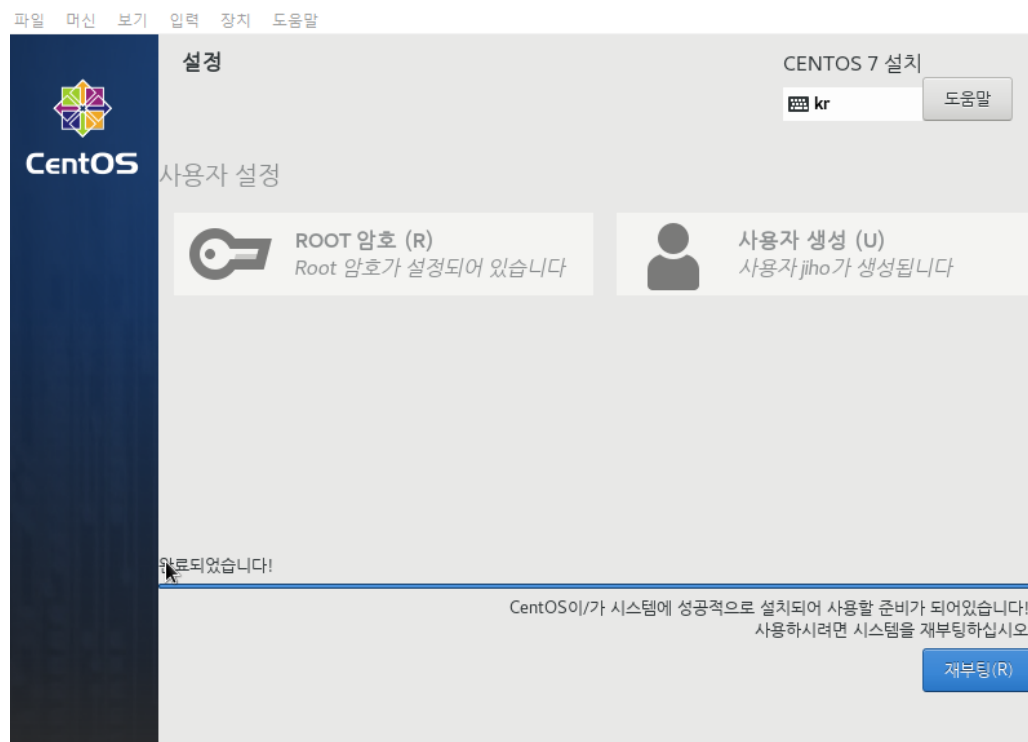


### 3) 네트워크 설정

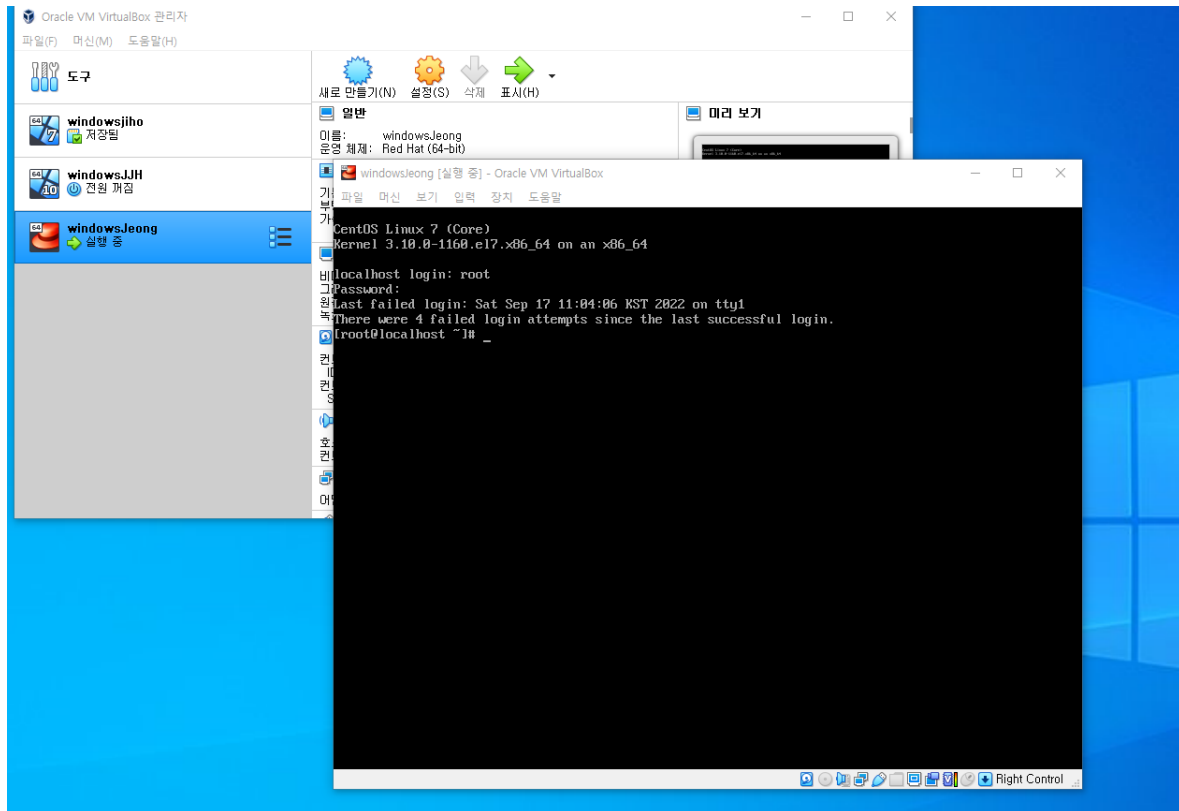


### 4) 설치 후 사용자 설정, 그리고 재부팅





## 5) 다시 실행 후 로그인(아래 사진은 로그인 된 상태이다)



## 6) 이제 명령어 입력이 가능하다

```
total 232
-rw-r--r--. 1 root root 281 Sep 17 11:42 ifcfg-enp0s3
-rw-r--r--. 1 root root 254 May 22 2020 ifcfg-lo
lrwxrwxrwx. 1 root root 24 Sep 17 2022 ifdown -> ../../usr/sbin/ifdown
-rwxr-xr-x. 1 root root 654 May 22 2020 ifdown-bnep
-rwxr-xr-x. 1 root root 6532 May 22 2020 ifdown-eth
-rwxr-xr-x. 1 root root 781 May 22 2020 ifdown-ippv
-rwxr-xr-x. 1 root root 4548 May 22 2020 ifdown-ipv6
lrwxrwxrwx. 1 root root 11 Sep 17 2022 ifdown-isdn -> ifdown-ippv
-rwxr-xr-x. 1 root root 2138 May 22 2020 ifdown-post
-rwxr-xr-x. 1 root root 1068 May 22 2020 ifdown-ppp
-rwxr-xr-x. 1 root root 878 May 22 2020 ifdown-routes
-rwxr-xr-x. 1 root root 1456 May 22 2020 ifdown-sit
-rwxr-xr-x. 1 root root 1621 Dec 9 2018 ifdown-Team
-rwxr-xr-x. 1 root root 1556 Dec 9 2018 ifdown-TeamPort
-rwxr-xr-x. 1 root root 1462 May 22 2020 ifdown-tunnel
lrwxrwxrwx. 1 root root 22 Sep 17 2022 ifup -> ../../usr/sbin/ifup
-rwxr-xr-x. 1 root root 12415 May 22 2020 ifup-aliases
-rwxr-xr-x. 1 root root 910 May 22 2020 ifup-bnep
-rwxr-xr-x. 1 root root 13758 May 22 2020 ifup-eth
-rwxr-xr-x. 1 root root 12075 May 22 2020 ifup-ippv
-rwxr-xr-x. 1 root root 11893 May 22 2020 ifup-ipv6
lrwxrwxrwx. 1 root root 9 Sep 17 2022 ifup-isdn -> ifup-ippv
-rwxr-xr-x. 1 root root 650 May 22 2020 ifup-plip
-rwxr-xr-x. 1 root root 1064 May 22 2020 ifup-plusb
-rwxr-xr-x. 1 root root 4997 May 22 2020 ifup-post
-rwxr-xr-x. 1 root root 4154 May 22 2020 ifup-ppp
-rwxr-xr-x. 1 root root 2001 May 22 2020 ifup-routes
-rwxr-xr-x. 1 root root 3303 May 22 2020 ifup-sit
-rwxr-xr-x. 1 root root 1755 Dec 9 2018 ifup-Team
-rwxr-xr-x. 1 root root 1876 Dec 9 2018 ifup-TeamPort
-rwxr-xr-x. 1 root root 2780 May 22 2020 ifup-tunnel
-rwxr-xr-x. 1 root root 1836 May 22 2020 ifup-wireless
-rwxr-xr-x. 1 root root 5419 May 22 2020 init.ipv6-global
-rw-r--r--. 1 root root 20678 May 22 2020 network-functions
-rw-r--r--. 1 root root 38988 May 22 2020 network-functions-ipv6
[root@localhost network-scripts]#
```

## - Ubuntu 설치

### 1) 우분투 공식홈페이지에서 다운로드

The screenshot shows the Ubuntu website's download section. At the top, there's a navigation bar with 'ubuntu' logo and links for Enterprise, Developer, Community, and Download. Below this, three main categories are highlighted: Ubuntu Desktop, Ubuntu Server, and Ubuntu for IoT. Each category has a brief description and a 'Get' button. Ubuntu Desktop offers 22.04 LTS. Ubuntu Server provides links for Mac and Windows, ARM, IBM Power, and s390x. Ubuntu for IoT lists various hardware platforms like Raspberry Pi, Intel IoT platforms, Intel NUC, KVM, Qualcomm Dragonboard 410c, Intel IEI TANK 870, AMD-Xilinx Evaluation kits & SOMs, and RISC-V platforms. At the bottom, there are sections for Tutorials, Read the Docs, and Other ways to download.

**Ubuntu Desktop**  
Download Ubuntu desktop and replace your current operating system whether it's Windows or Mac OS, or, run Ubuntu alongside it.  
**22.04 LTS**

**Ubuntu Server**  
The most popular server Linux in the cloud and data centre, you can rely on Ubuntu Server and its five years of guaranteed free upgrades.  
**Get Ubuntu Server**  
Mac and Windows  
ARM  
IBM Power  
s390x

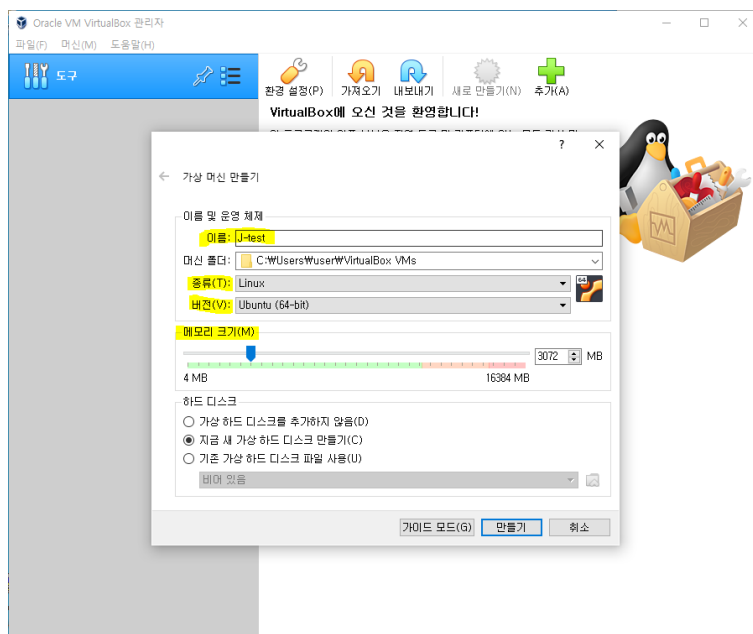
**Ubuntu for IoT**  
Are you a developer who wants to try snappy Ubuntu Core or classic Ubuntu on an IoT board?  
Raspberry Pi  
Intel IoT platforms  
Intel NUC  
KVM  
Qualcomm Dragonboard 410c  
Intel IEI TANK 870  
AMD-Xilinx Evaluation kits & SOMs  
RISC-V platforms

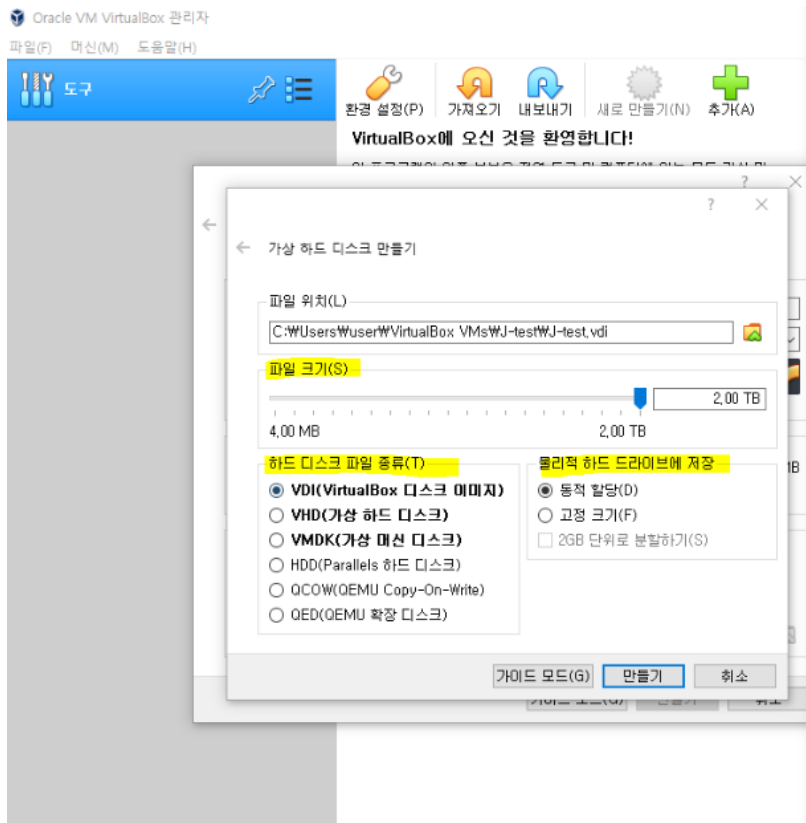
**TUTORIALS**  
If you are already running Ubuntu - you can [upgrade](#) with the Software Updater  
Burn a DVD on [Ubuntu](#), [macOS](#), or

**READ THE DOCS**  
Read the official docs for [Ubuntu Desktop](#), [Ubuntu Server](#), and [Ubuntu Core](#)

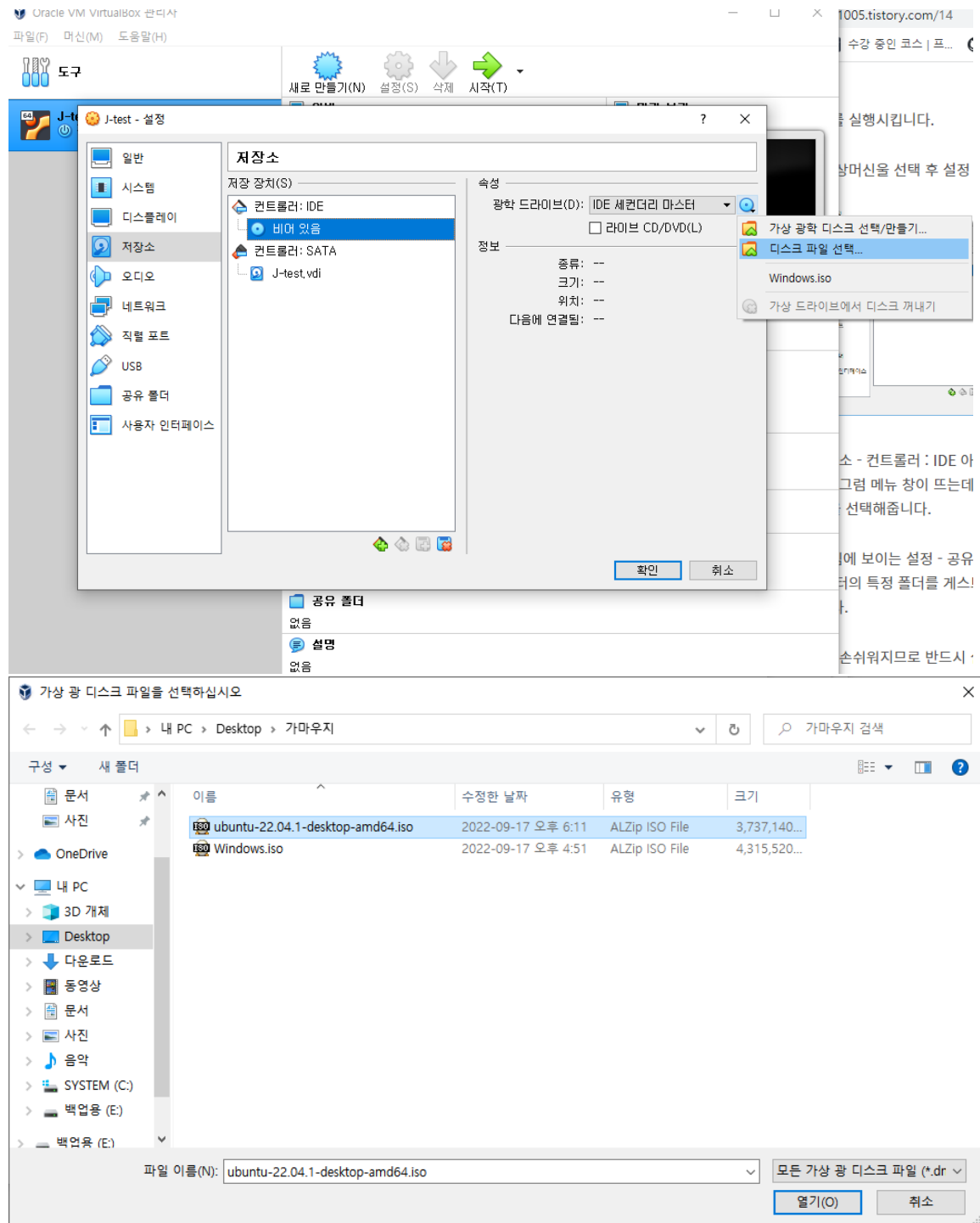
**OTHER WAYS TO DOWNLOAD**  
Ubuntu is available via [BitTorrents](#) and via a minimal [network installer](#) that allows you to customise what is installed, such as additional languages. You can also find [older](#)

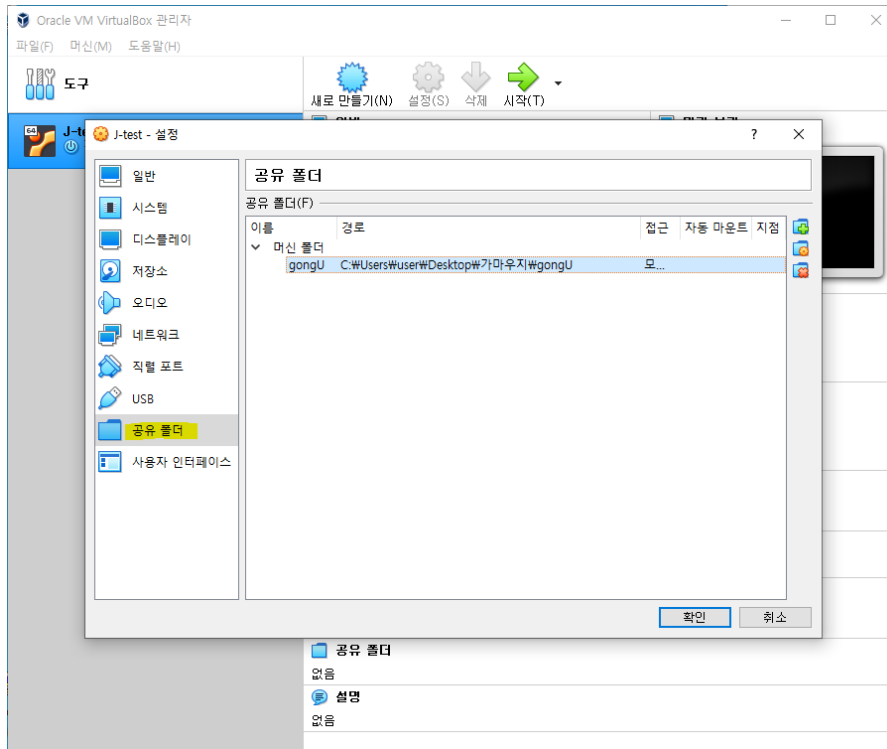
### 2) 가상 머신 만들기



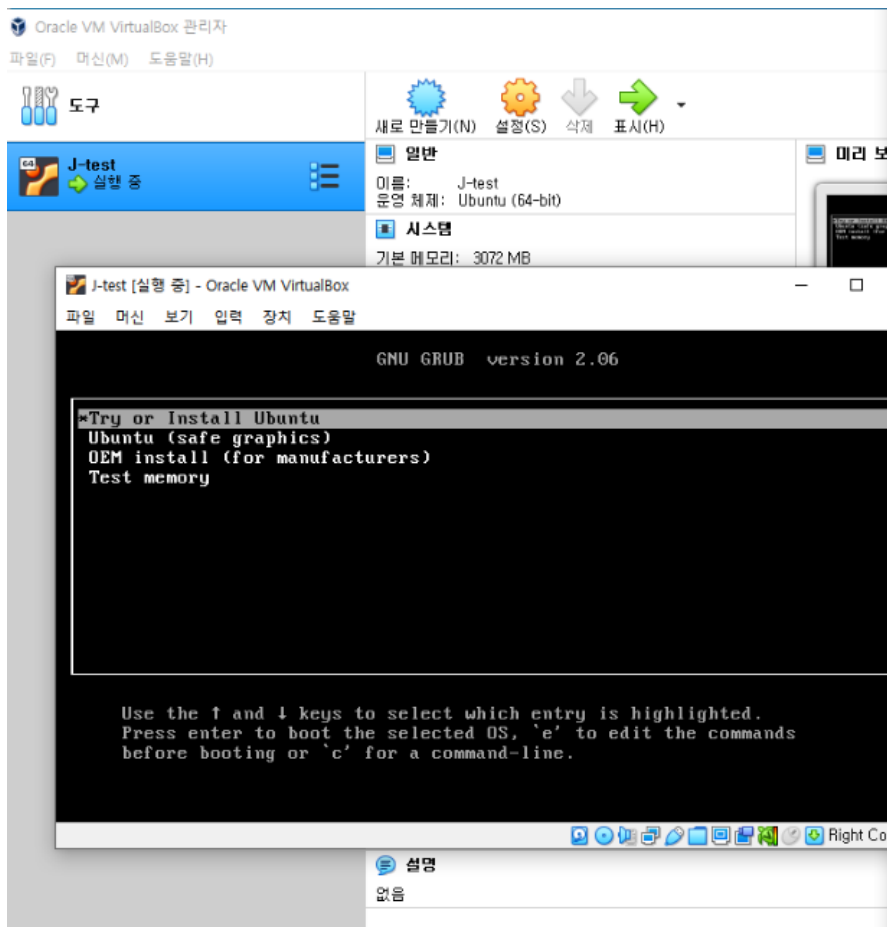


### 3) iso 및 공유폴더 설정



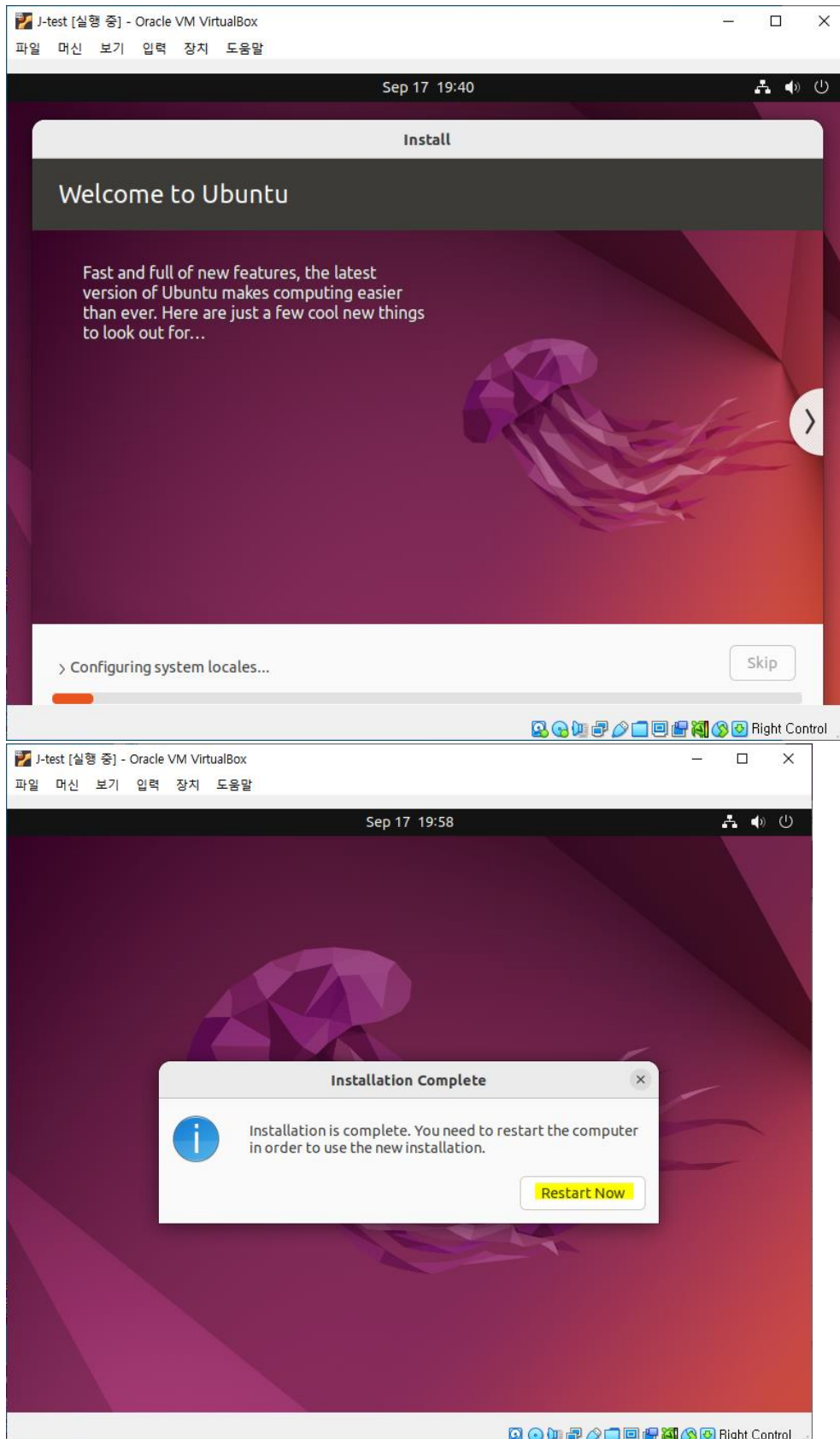


#### 4) 실행 후 try or install ubuntu





5) install 작업을 수행한다. 완료되면 재시작한다.



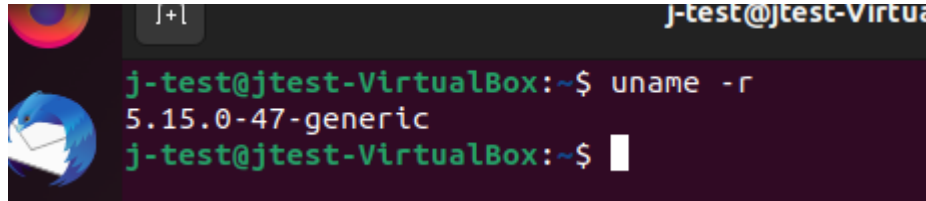
## - CentOS vs UBUNTU

### CentOS 와 ubuntu 의 차이점

- 1) CentOS 는 **RedHat** 기반 vs ubuntu 는 **Debian** 기반
- 2) CentOS 는 **서버용**으로 리눅스를 사용할 경우  
사용 vs ubuntu 는 **데스크탑용**으로 주로 쓰임
- 3) CentOS 는 **안정적** vs ubuntu 는 **CentOS 에 비해 불안정하고, 안정화 되기까지 시간이 걸림.**
- 4) CentOS 는 **.rpm, yum** 패키지 관리 vs ubuntu 는 **.deb** 패키지와 **apt-get** 패키지로 패키지 관리
- 5) CentOS 는 최신 업데이트에 보수적 vs ubuntu 는 배포 주기가 짧은 편.  
**CentOS 보다 새로운 소프트웨어 및 레포지토리에 더 빨리 접근 가능**
- 6) CentOS 보안은 양호하지만, 추가 구성이 필요 vs ubuntu 보안은 강함
- 7) CentOS 클라우드인터페이스는 **OpenStack, OpenNebula, CloudStack** vs ubuntu 클라우드인터페이스는 **OpenStack**
- 8) CentOS 호스팅 시장 점유율은 17.3 % 정도 vs ubuntu 호스팅 시장 점유율은 37.8 % 정도

### 3. 커널 업데이트 & 다운그레이드

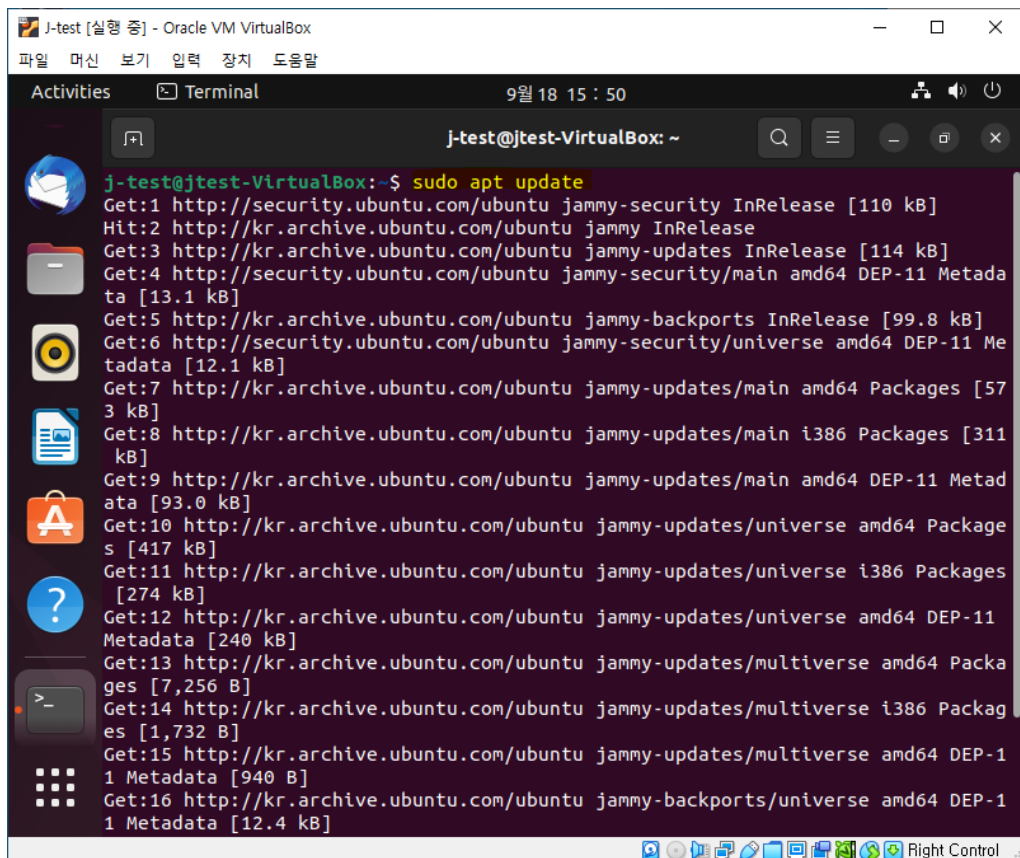
- 현재 커널 버전을 확인한다(`uname -r`)



```
j-test@jtest-VirtualBox:~$ uname -r
5.15.0-47-generic
j-test@jtest-VirtualBox:~$
```

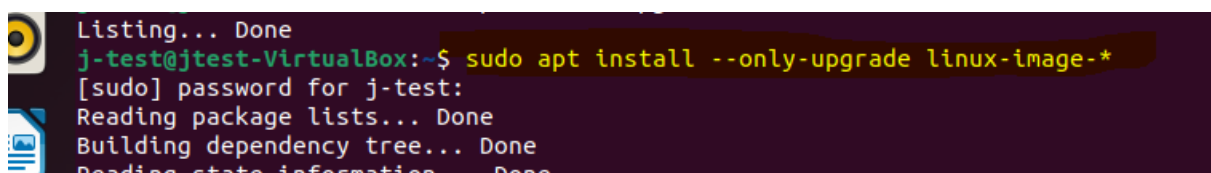
- 시스템 내 전체 패키지 업그레이드를 통한 커널 업그레이드

(`sudo apt update`)



```
j-test@jtest-VirtualBox:~$ sudo apt update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://kr.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://kr.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [13.1 kB]
Get:5 http://kr.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [12.1 kB]
Get:7 http://kr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [573 kB]
Get:8 http://kr.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [311 kB]
Get:9 http://kr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [93.0 kB]
Get:10 http://kr.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [417 kB]
Get:11 http://kr.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [274 kB]
Get:12 http://kr.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [240 kB]
Get:13 http://kr.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [7,256 B]
Get:14 http://kr.archive.ubuntu.com/ubuntu jammy-updates/multiverse i386 Packages [1,732 B]
Get:15 http://kr.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:16 http://kr.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [12.4 kB]
```

- 커널만 업데이트 할 수도 있다.



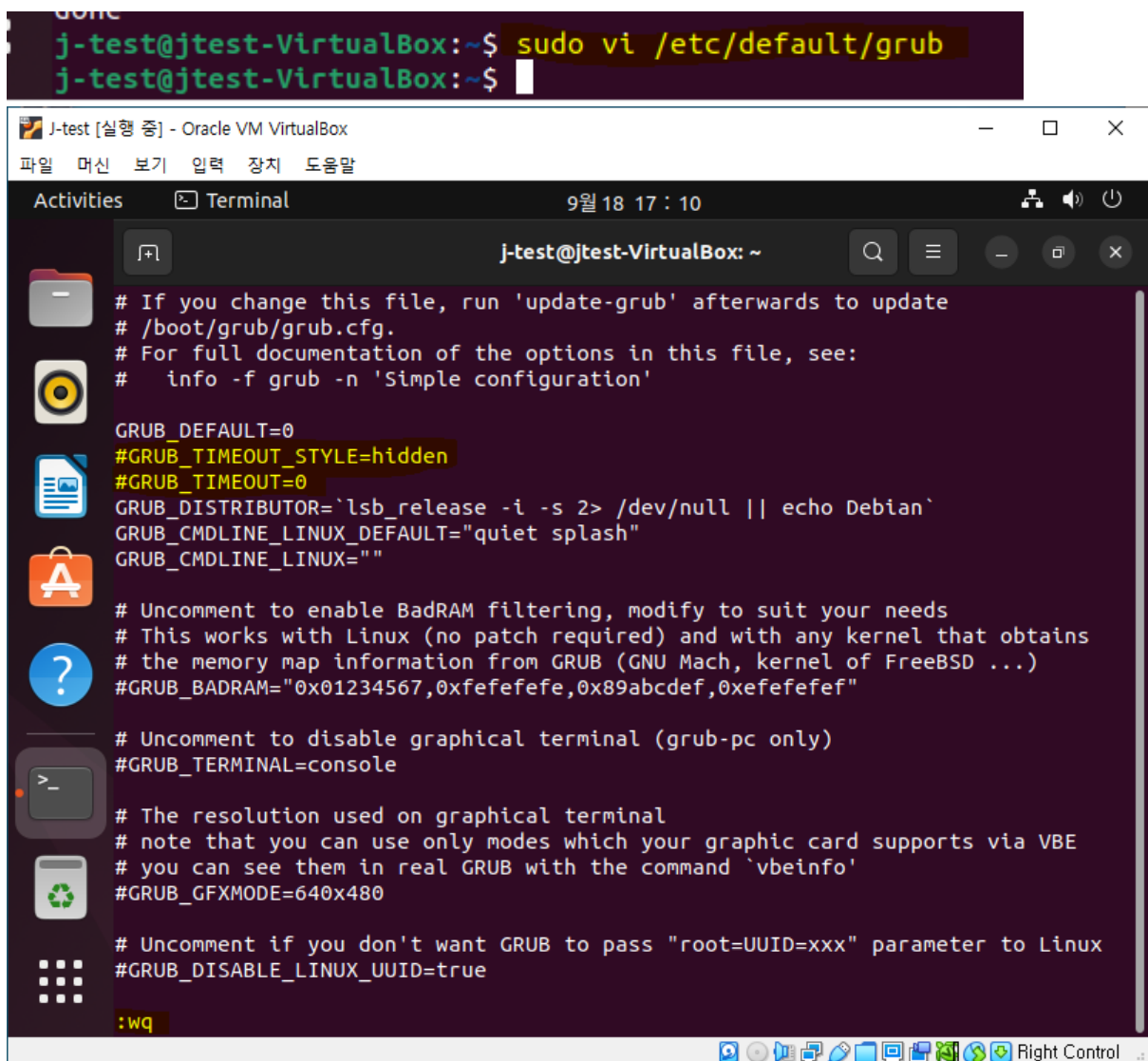
```
j-test@jtest-VirtualBox:~$ sudo apt install --only-upgrade linux-image-*
[sudo] password for j-test:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

- 리부팅 후 확인하면 업데이트가 된 것을 확인할 수 있다.

## - Kernal 버전 변경

1. 부팅할 때마다 부트로더인 Grub2 화면이 보이게 만든다면, 원하는 커널 버전을 선택하여 부팅할 수 있다.

2. 부팅 때 Grub2 를 실행하기 위해 Grub2 의 설정을 변경해주자.



```

j-test@jtest-VirtualBox:~$ sudo vi /etc/default/grub
j-test@jtest-VirtualBox:~$

# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
#GRUB_TIMEOUT_STYLE=hidden
#GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

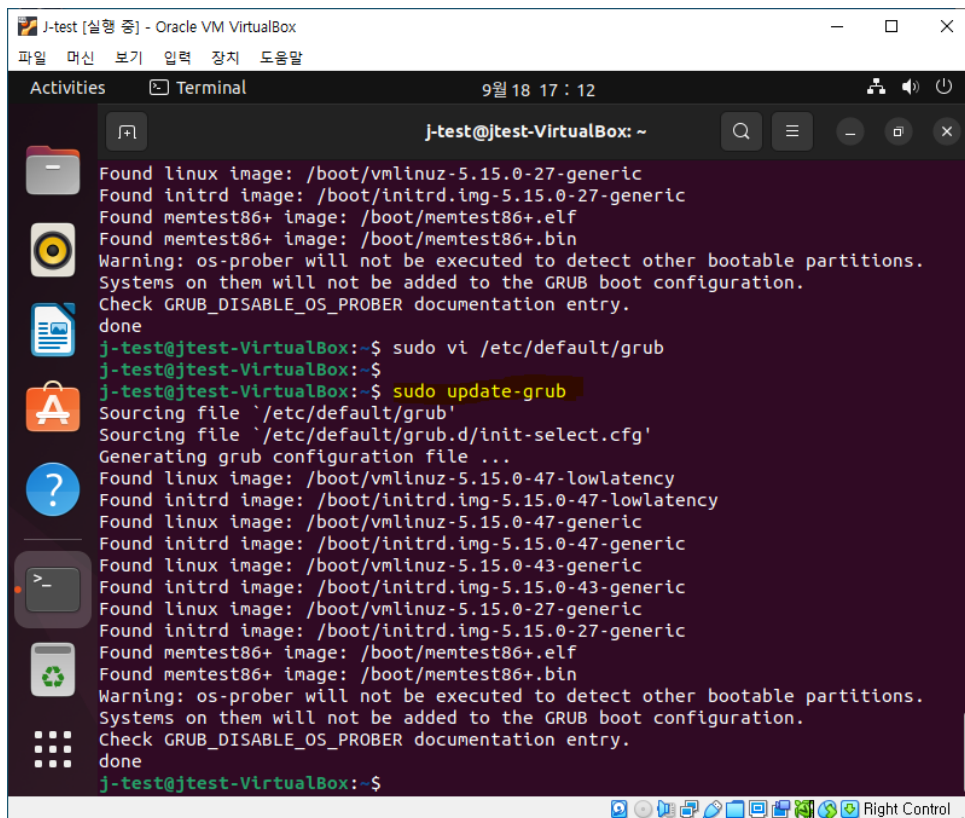
# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

:wq
  
```

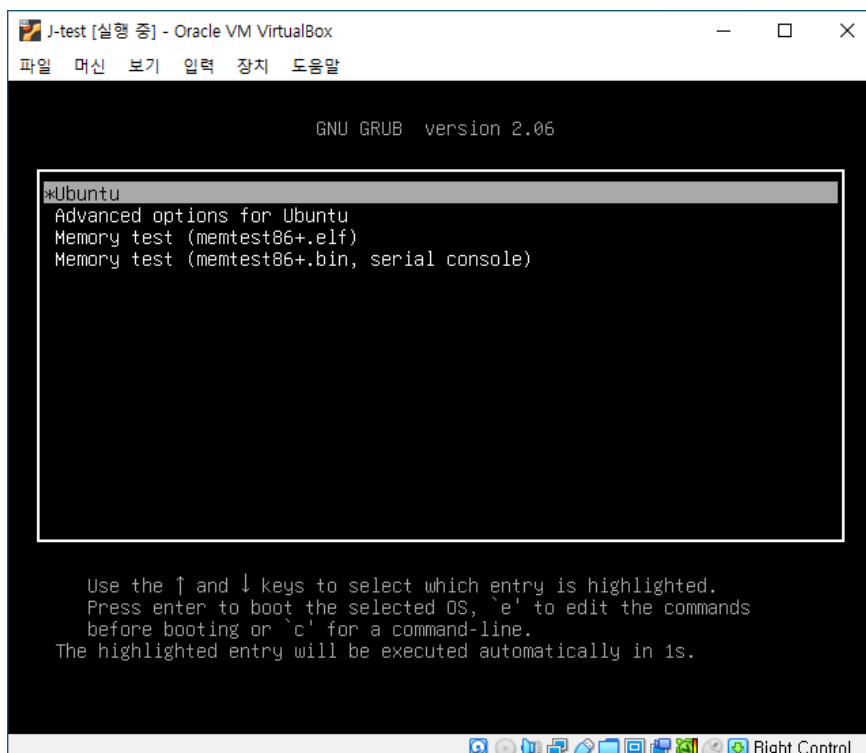
3. GRUB\_TIMEOUT\_STYLE, GRUB\_TIMEOUT을 주석처리 해준다. 그리고 저장.



```

j-test@jtest-VirtualBox: ~
Found linux image: /boot/vmlinuz-5.15.0-27-generic
Found initrd image: /boot/initrd.img-5.15.0-27-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
j-test@jtest-VirtualBox:~$ sudo vi /etc/default/grub
j-test@jtest-VirtualBox:~$
j-test@jtest-VirtualBox:~$ sudo update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.15.0-47-lowlatency
Found initrd image: /boot/initrd.img-5.15.0-47-lowlatency
Found linux image: /boot/vmlinuz-5.15.0-47-generic
Found initrd image: /boot/initrd.img-5.15.0-47-generic
Found linux image: /boot/vmlinuz-5.15.0-43-generic
Found initrd image: /boot/initrd.img-5.15.0-43-generic
Found linux image: /boot/vmlinuz-5.15.0-27-generic
Found initrd image: /boot/initrd.img-5.15.0-27-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
j-test@jtest-VirtualBox:~$
  
```

4. 적용을 해준다.



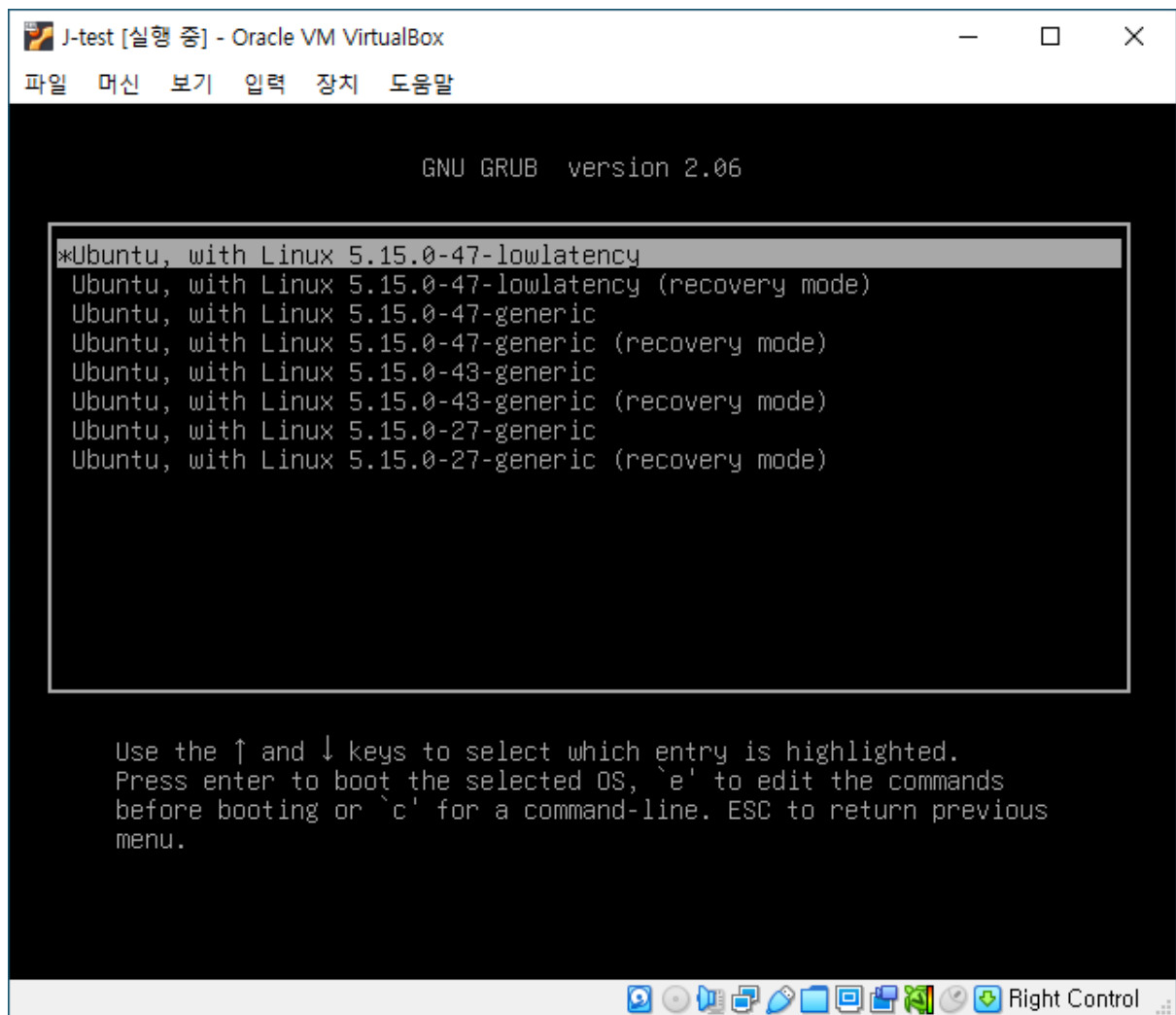
```

GNU GRUB version 2.06

*Ubuntu
Advanced options for Ubuntu
Memory test (memtest86+.elf)
Memory test (memtest86+.bin, serial console)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.
The highlighted entry will be executed automatically in 1s.
  
```

## 5. Grub2의 화면



## 4. CentOS bonding 구성 및 0~6번의 옵션에 대해

### - Centos bonding 구성

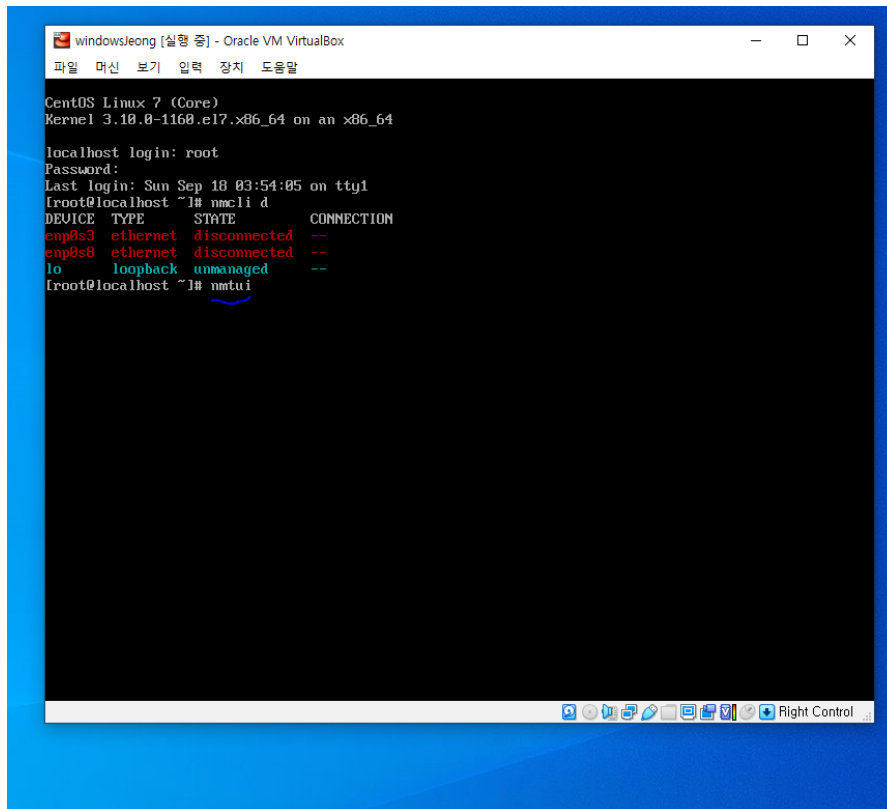
1) 현재 설정 가능한 디바이스를 확인(nmcli d 명령어 사용)

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

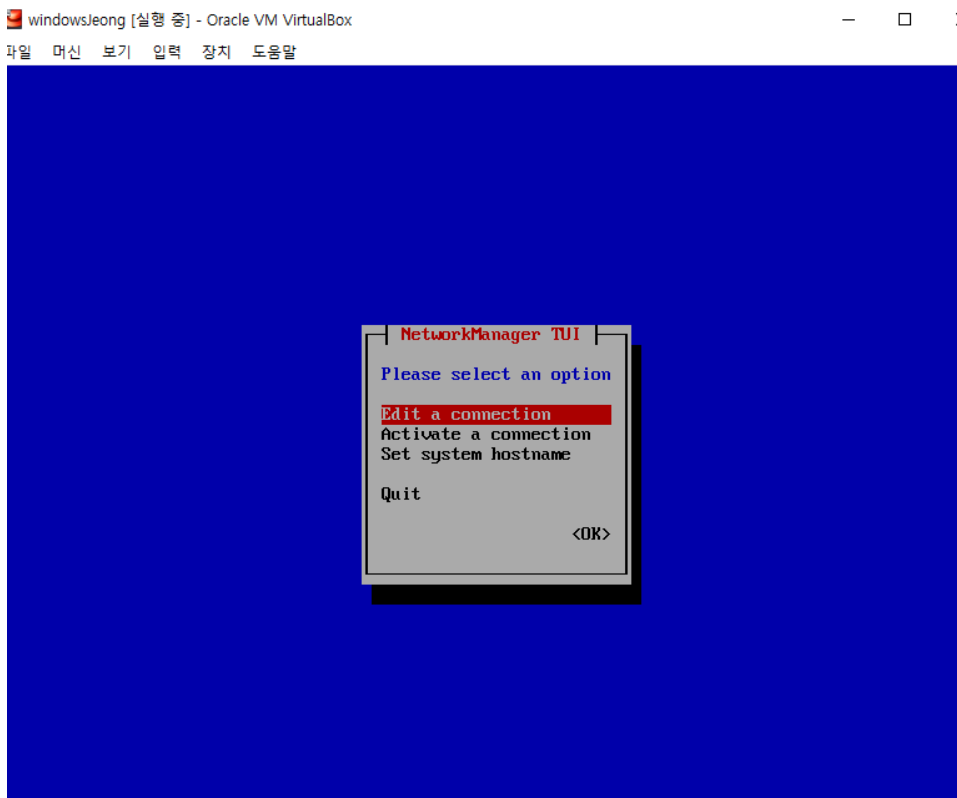
localhost login: root
Password:
Last login: Sun Sep 18 03:54:05 on tty1
[root@localhost ~]# nmcli d
DEVICE    TYPE      STATE      CONNECTION
enp0s3    ethernet  disconnected --
enp0s8    ethernet  disconnected --
lo        loopback  unmanaged  --
[root@localhost ~]#
```

-> enp0s3와 enp0s8가 있으므로 이 둘을 Bonding 하자

## 2) nmtui 명령어 입력

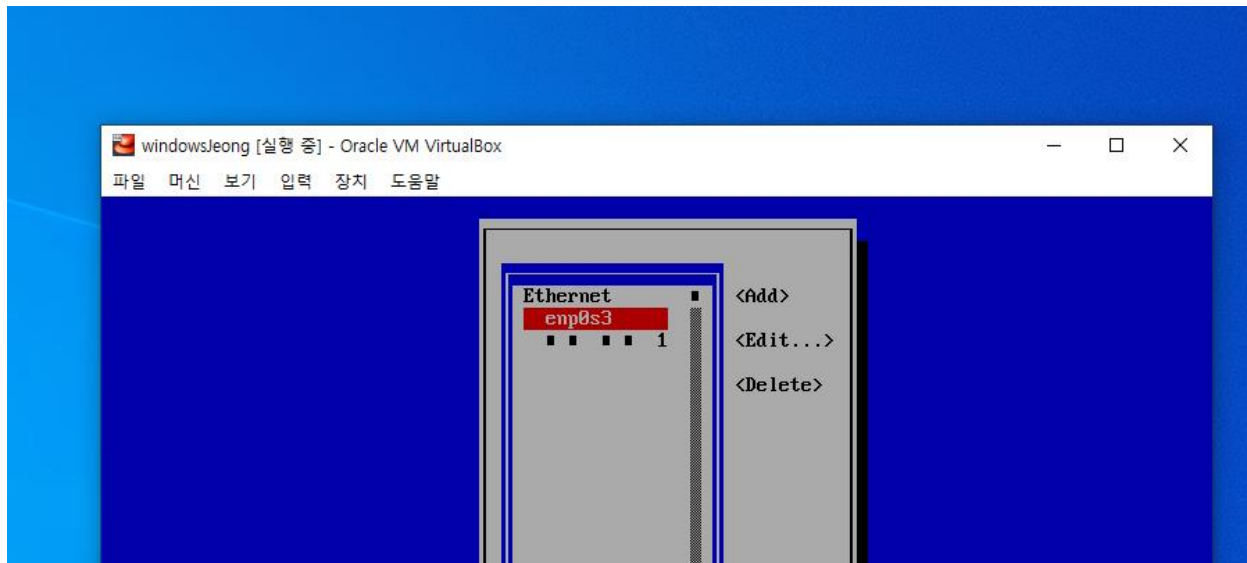


## 3) Edit a connection 선택

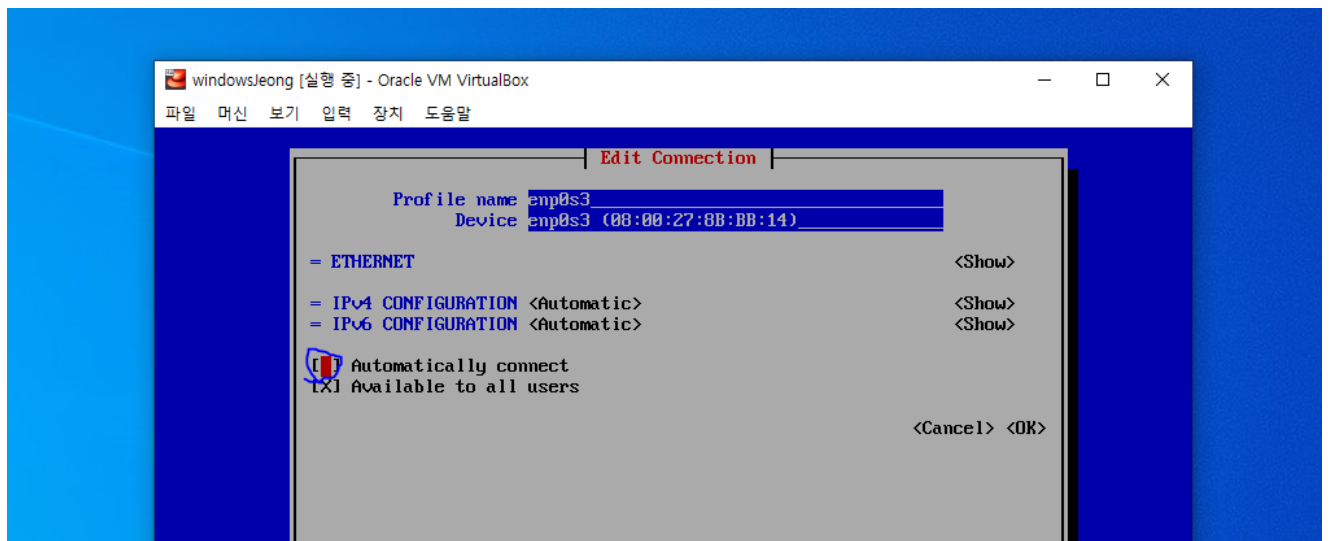




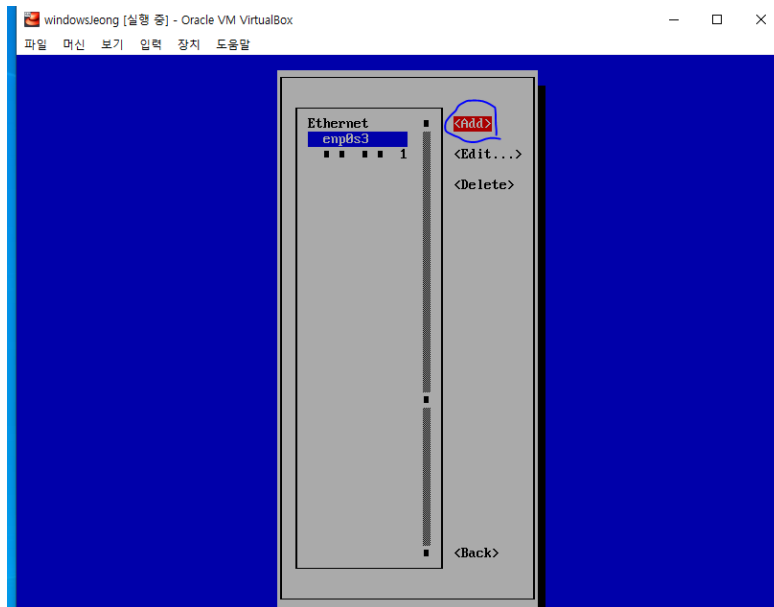
4) 네트워크 연결에 사용 중이던 eth를 선택한다(enp0s3).



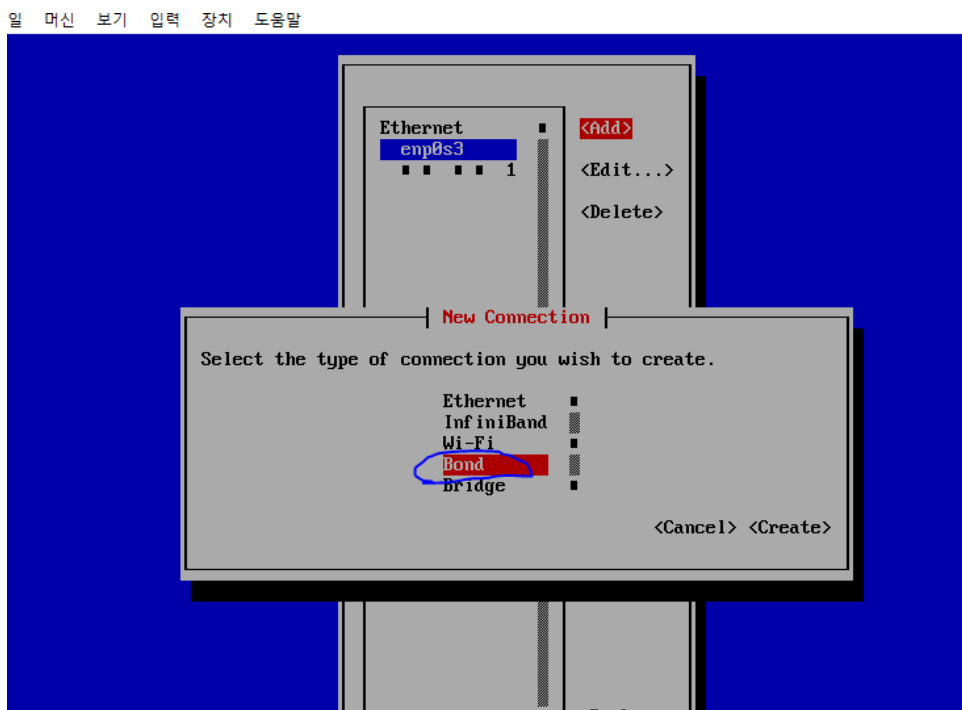
5) Automatically connect를 체크 해제하고 ok를 눌러 완료



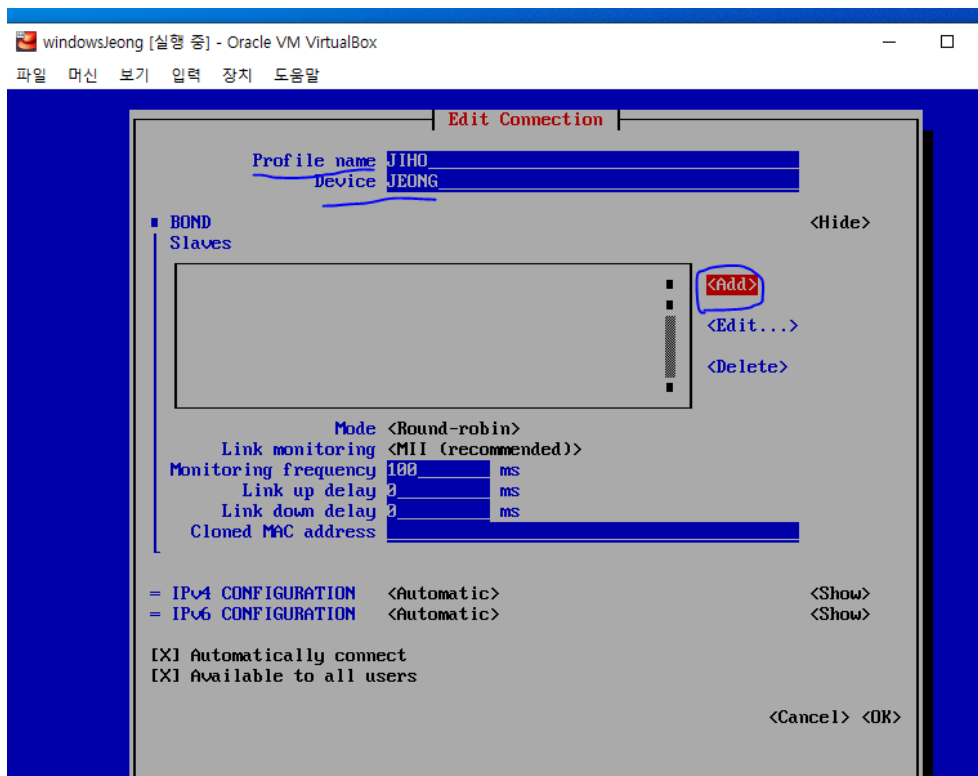
## 6) Add 선택



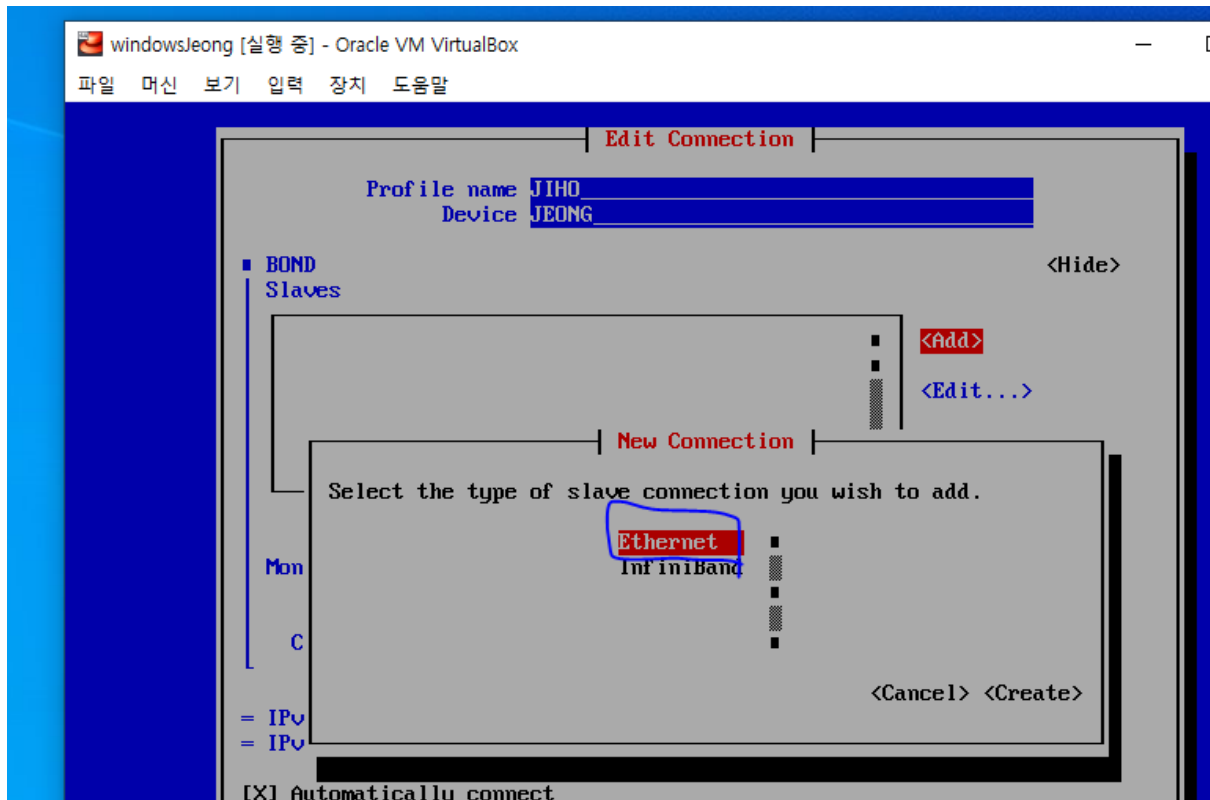
## 7) Bond 선택



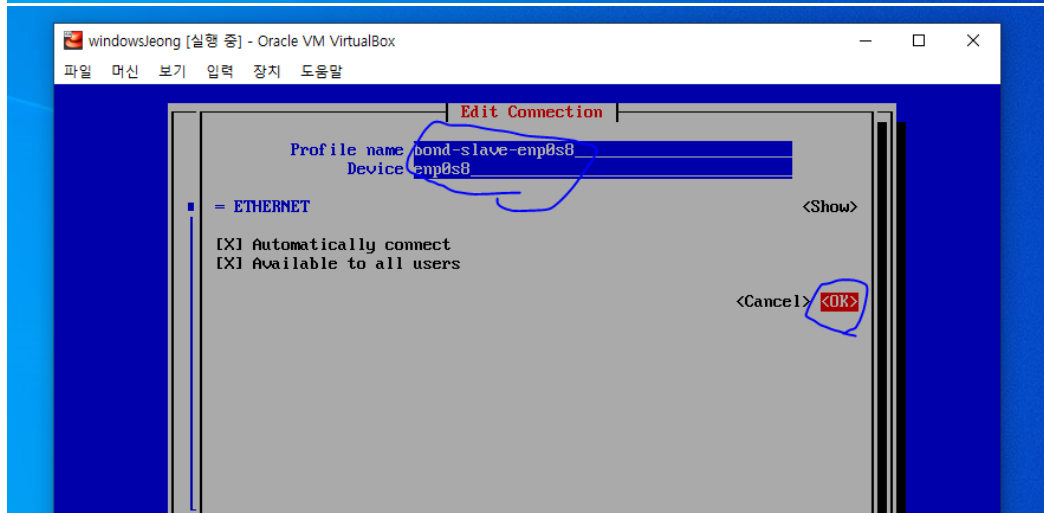
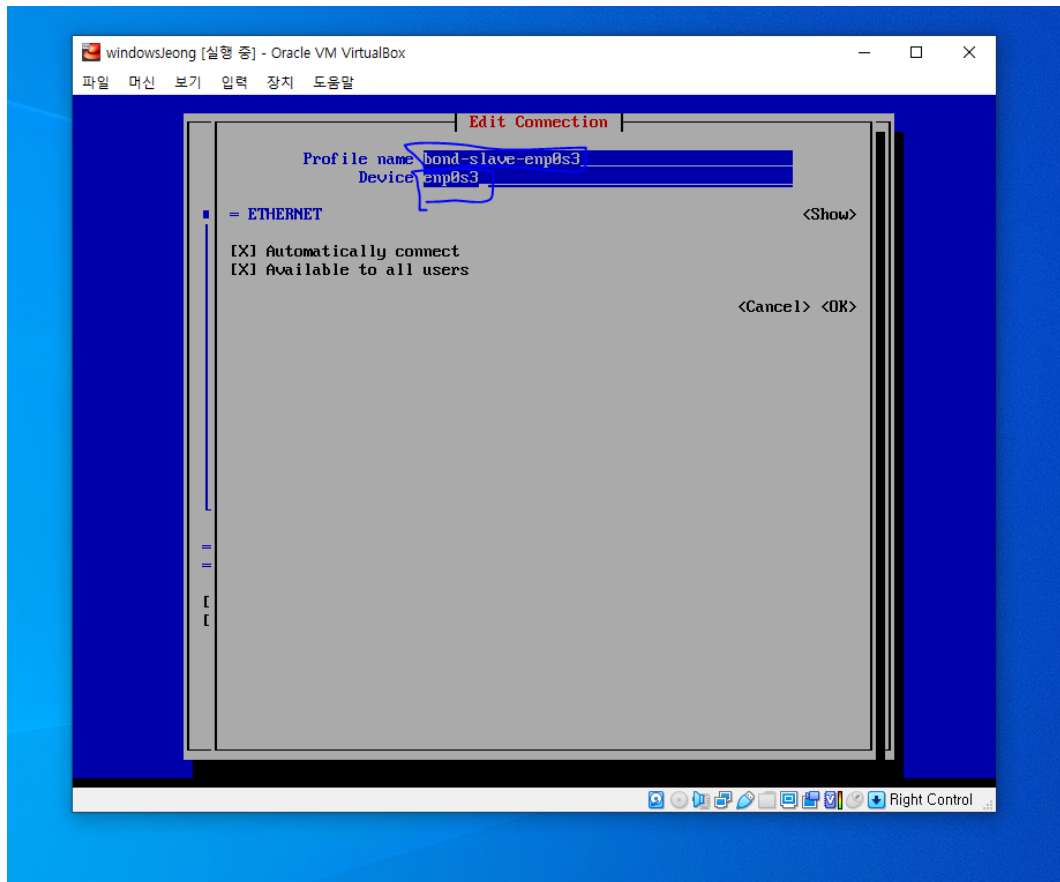
8) 가상 네트워크 bond의 Profile name과 Device를 입력 후, BOND-Slaves-Add를 선택



9) Ethernet 선택

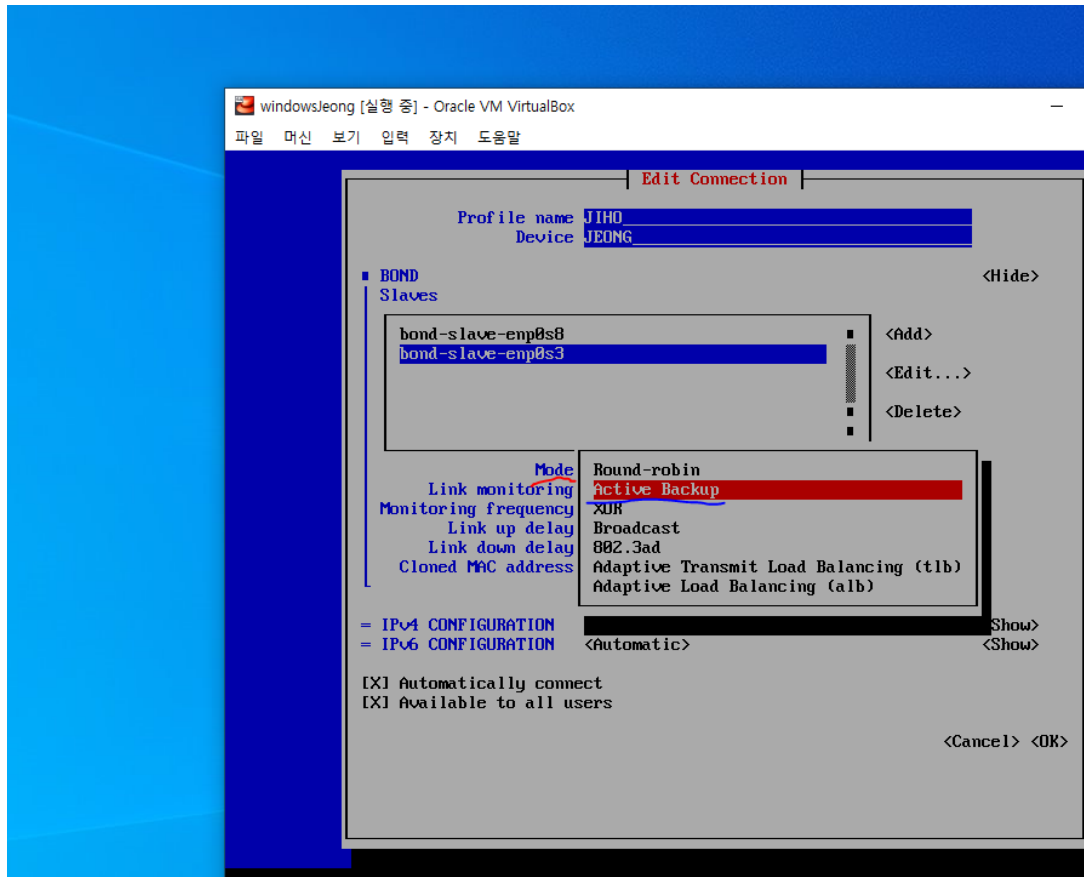


# 10) Bonding을 구성할 네트워크 인터페이스 enp0s3와 enp0s8을 추가

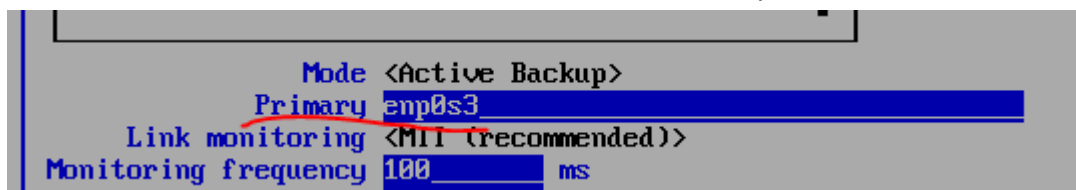


11) bonding 모드는 Active Backup 사용

(Active Backup: 물리적 환경과 가상 환경의 백업 작업을 통합하고, 파일 또는 VM을 신속하게 복원할 수 있다)



12) Active로 사용할 네트워크 인터페이스 선택 (enp0s3)



13) IPv4 CONFIGURATION을 Manual로 설정한 후, show 버튼을 누른다.

The screenshot shows the IPv4 CONFIGURATION menu. The Mode dropdown is set to 'Manual'. Other options include 'Active Backup', 'enp0s3', 'MII (recommended)', '100 ms', '0 ms', '0 ms', and 'Cloned MAC address'. There are buttons for '<Show>' and '<Cancel> <OK>'. The menu also includes options for IPv6 CONFIGURATION and checkboxes for 'Automatically connect' and 'Available to all users'.

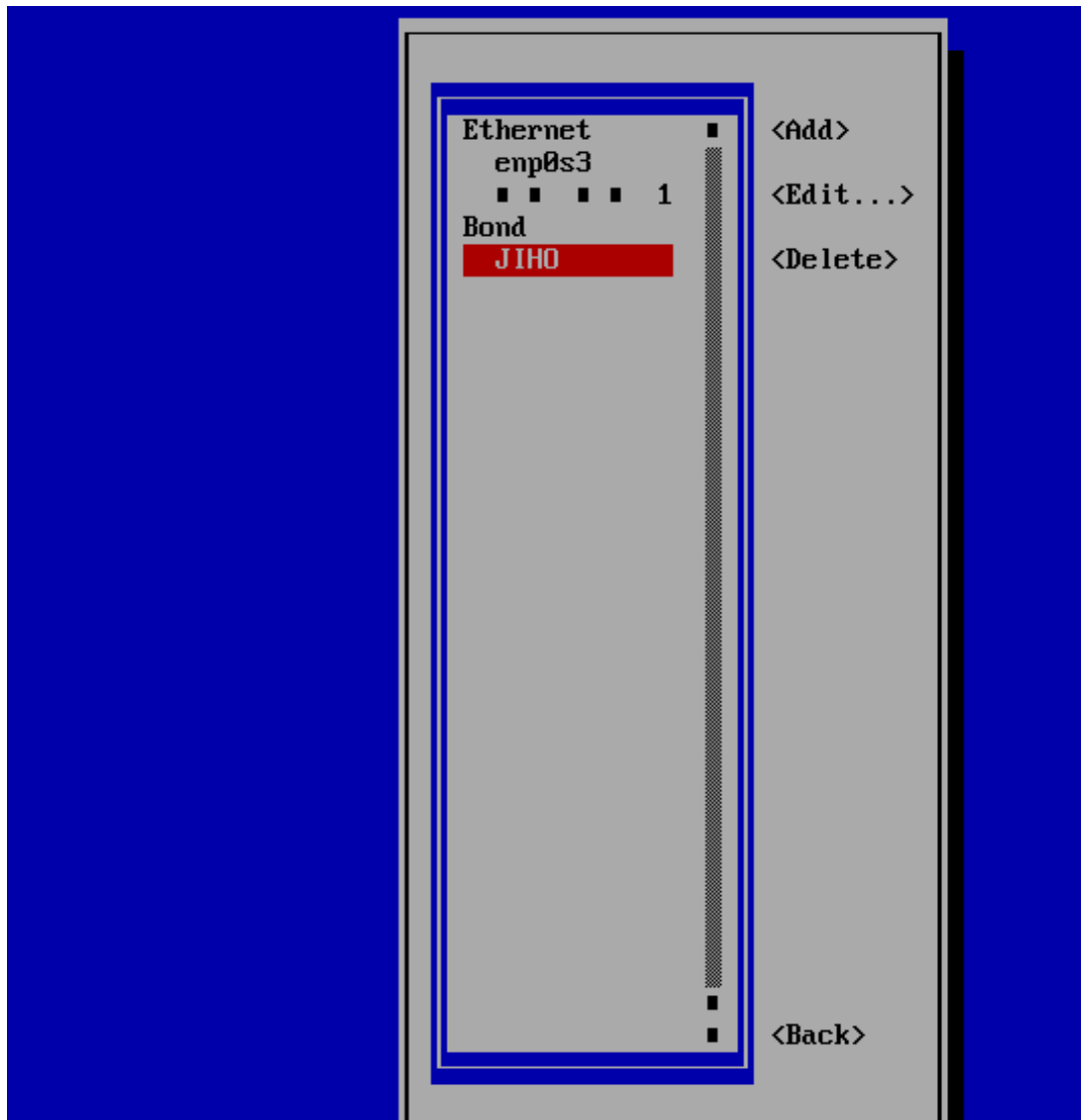
14) Address, Gateway, DNS Servers 입력

The screenshot shows the IPv4 CONFIGURATION menu with the Mode set to 'Manual'. The input fields for 'Addresses', 'Gateway', 'DNS servers', and 'Search domains' are visible. The 'Addresses' field contains '10.188.191.114'. The 'Gateway' field contains '10.188.191.1'. The 'DNS servers' field contains '168.188.1.1' and '168.188.1.2'. There are buttons for '<Remove>', '<Add...>', and '<Cancel> <OK>'.

15) OK 눌러 완료

The screenshot shows the IPv4 CONFIGURATION menu with the Mode set to 'Manual'. The input fields for 'Addresses', 'Gateway', 'DNS servers', and 'Search domains' are visible. The 'Addresses' field contains '10.188.191.114'. The 'Gateway' field contains '10.188.191.1'. The 'DNS servers' field contains '168.188.1.1' and '168.188.1.2'. There are buttons for '<Remove>', '<Add...>', and '<Cancel> <OK>'. The menu also includes options for IPv6 CONFIGURATION and checkboxes for 'Automatically connect' and 'Available to all users'.

16) JIHO라는 인터페이스가 생성되었다.



## - Bonding 옵션 (0~6번 옵션)

### - mode 0

1) 명칭: **Balance-RR**

2) 특징: **Load Balancing(Round-Robin)** / 모든 슬레이브 NIC(Network Interface Controller. 컴퓨터를 네트워크에 연결하여 통신하기 위해 사용하는 하드웨어 장치)를 포함 / 순차적으로 데이터 패킷 전송, 전송 패킷만 부하 분산 / 스위치 설정이 필요함 / (**Round-Robin** 이란 프로세스들 사이에 우선순위를 두지 않고 **순서대로 시간단위로 CPU 를 할당하는 방식**을 말한다.) / (패킷: 컴퓨터 네트워크가 전달하는 데이터의 형식화된 블록)

### - mode 1

1) 명칭: Active-Backup

2) 특징: 하나의 NIC 장치만 Active, Active Interface 만 패킷 송수신 / 송수신 패킷 부하분산이 없음 / **연결 장비 설정 필요 x**

### - mode 2

1) 명칭: Balance-Xor

2) 특징: [출발지 MAC + 도착지 MAC + module 슬레이브 1 개]에 기초하여 전송 / 송신 패킷만 분산 / 스위치 설정 필요함

### - mode 3

1) 명칭: Broadcast

2) 특징: 송수신 패킷을 복사하여 모든 Ethernet Device 로 동일한 패킷 전송 / **연결 장비 설정 필요 x**



- mode 4
  - 1) 명칭: 802.3ad
  - 2) 특징: 송,수신 패킷 분산 / 스위치 설정 필요함
  
- mode 5:
  - 1) 명칭: Balance-tlb
  - 2) 특징: 송신 패킷 부하분산 / 스위치 설정 필요 x
  
- mode 6:
  - 1) 명칭: Balance-alb
  - 2) 특징: 5 번 옵션 기능 + 송수신 패킷 부하분산 / 스위치 설정 필요 x

#### -모드 활용

- 서버 본딩시 스위치 지원이 필요한 경우: mode 0,2,4
- 서버 본딩시 스위치 지원이 필요 없는 경우: mode 3,5,6
- 평균적으로 mode 0,1,4 많이 사용
- 서버와 연결 될 상대 장비에 LACP(Link aggregation Contorol Protocol. 여러개의 물리적 포트들을 묶어서 하나의 논리적인 포트로 동작하게 만드는 기술)가 지원이 되지 않는 경우에는 mode 6 사용