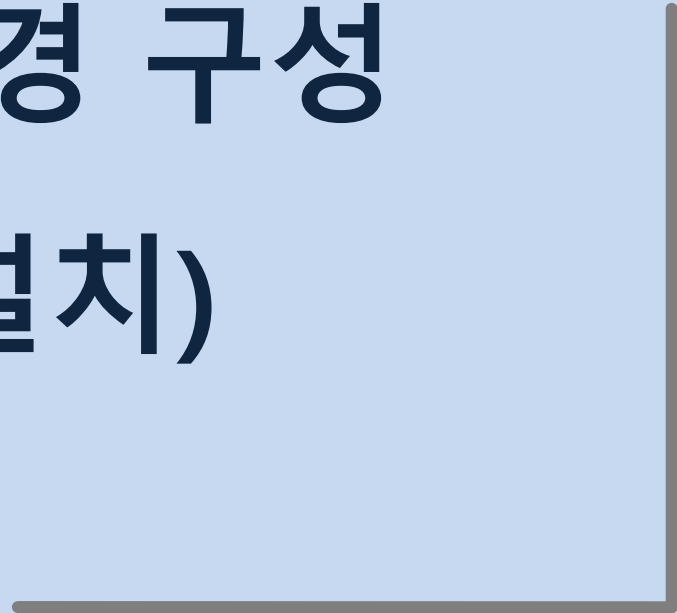




개발 환경 구성 (PC설치)



개발 환경

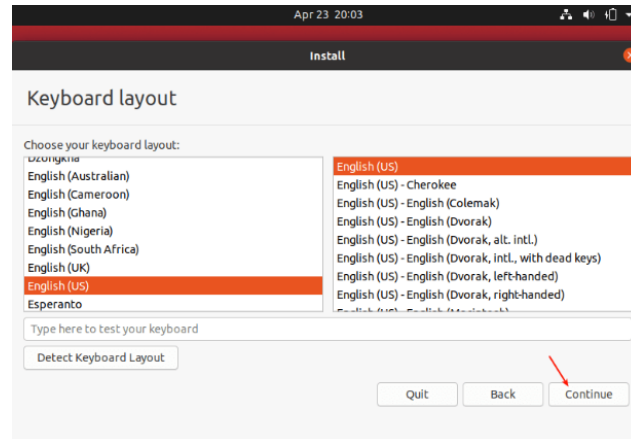
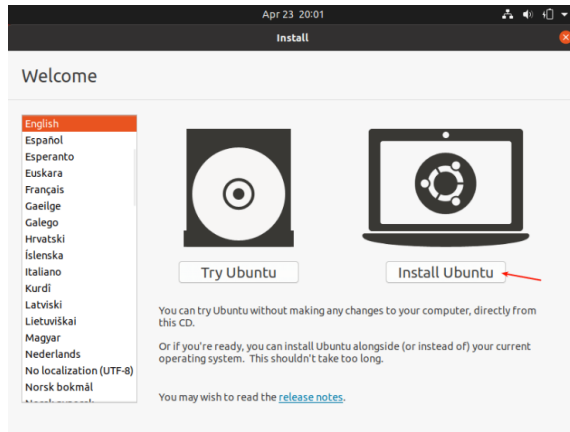
구분	사항
기본 운영체제	Ubuntu 20.04.6 LTS
로봇 운영체제	ROS2 Foxy Fitzroy
컴퓨터 아키텍처	amd64
통합 개발환경(IDE)	Visual Studio Code
프로그래밍 언어	python 3
시뮬레이터	Gazebo 11
DDS	Fast DDS
기타	CMake, Qt, OpenCV

50GB이상 저장공간 필요

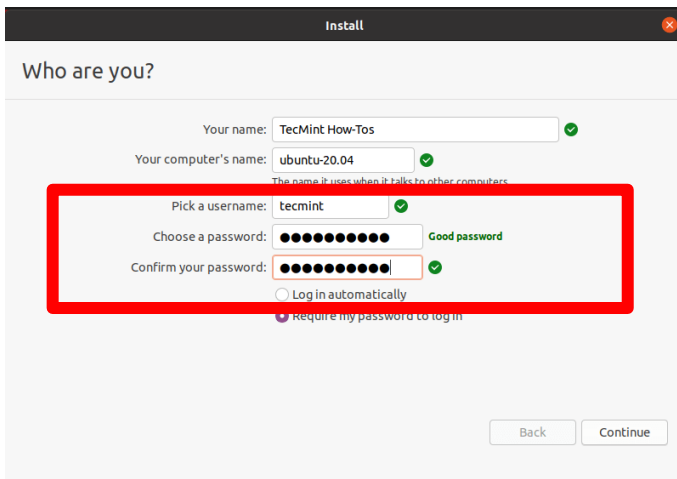
- **50GB 이상의 여유 저장공간 필요 - 리눅스 설치 공간**
 - 별도 하드 디스크를 추가함
 - 기존 하드 디스크의 파티션을 나눔
 - 외장 하드 디스크(SSD)를 추가함(외장SSD를 통한 Ubuntu 실행은 설치를 진행한 노트북에서만 할 수 있음)
- 파티션 분배 시 Ubuntu환경에 50GB이상 용량을 확보해야 함.
- 개인 노트북의 하드디스크 용량이 부족한 경우, **외장SSD에 Ubuntu를 설치해도 됨.**
- 외장SSD에 Ubuntu 설치를 진행한 후, 노트북 부팅 시 Ubuntu를 선택하시면 됩니다.
- 테스트 결과, **외장SSD를 통한 Ubuntu 실행은 설치를 진행한 노트북에서만 할 수 있습니다.**
- (즉, 설치를 진행한 노트북이 아닌 다른 노트북에 외장SSD를 연결하여 Ubuntu를 실행할 수 없습니다.)

Ubuntu 계정

- Welcome -> 언어는 English로 설정



- Who are you? -> Pick a username 설정



✓ Root계정 꼭 기억하기

✓ 성명이니셜+학번

✓ ex. 이름이 '김소희'이고, 학번이 '3717'인 경우 → **ksh3717**

✓ 비번은 **sshs1sshs!** 로 통일

설치 참고 사이트

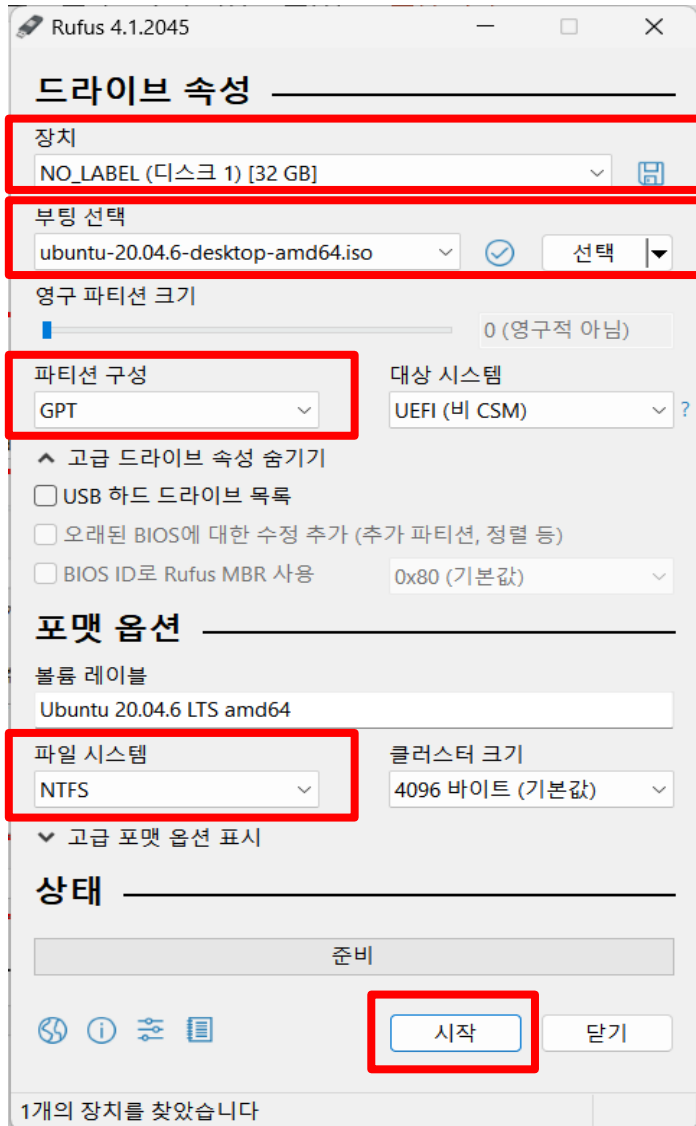
- Ubuntu 부팅디스크 만들기
 - <https://ubuntu.com/tutorials/create-a-usb-stick-on-windows#1-overview>
- Ubuntu 듀얼부팅 설치
 - <https://youtu.be/x7tpah6Tiqw>
- ROS 2 foxy 설치
 - <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/#pc-setup>
 - 링크 접속 후 버전인  를 반드시 선택

개발 환경 구성

- Ubuntu 20.04.6 LTS 설치
- 개발 프로그램 설치
 - Terminator
 - VScode : text editor
- 추가 설정
 - 한영키 설정
 - 사용자 계정 설정
- ROS2 foxy 설치
- ROS2 추가 프로그램 설치
 - Gazebo11
 - Cartographer
 - Navigation2
- 개발 환경 설정
 - Colcon build system(collective construction)
 - Workspace(이후 ws)
 - Bash 설정

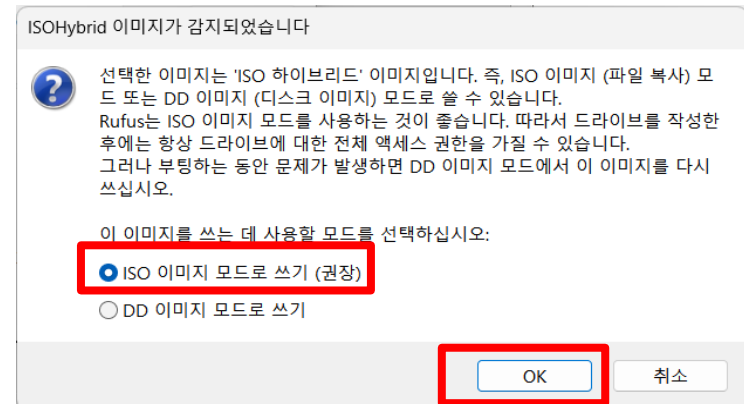
Ubuntu 20.04.6 LTS
ROS 2 foxy

Ubuntu 부팅디스크 생성



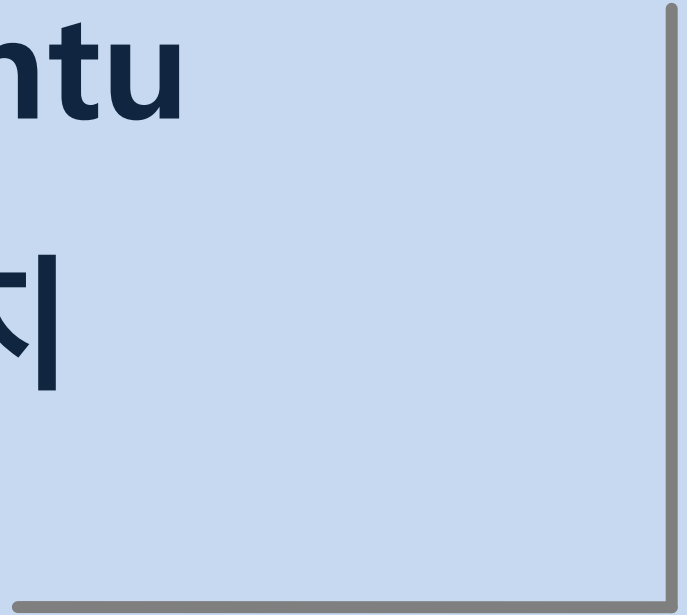
준비물: USB, Rufus 실행 파일, Ubuntu iso 파일

1. 장치 : USB 드라이브 선택
 - 설치 USB로 만들 USB를 선택
2. 부팅 선택 : 부팅 이미지 파일 선택
 - [선택]버튼 클릭 후 우분투 iso 이미지파일 선택
3. 파티션 구성 : GPT 선택
 - GPT 파티션 방식의 UEFI 시스템 방식으로 설정해야 함
4. 파일 시스템 : NTFS 선택.
5. [시작] 클릭
6. 팝업창 열린 경우 - ISO 이미지 모드로 쓰기 선택





Ubuntu 설치



BIOS 설정-Boot

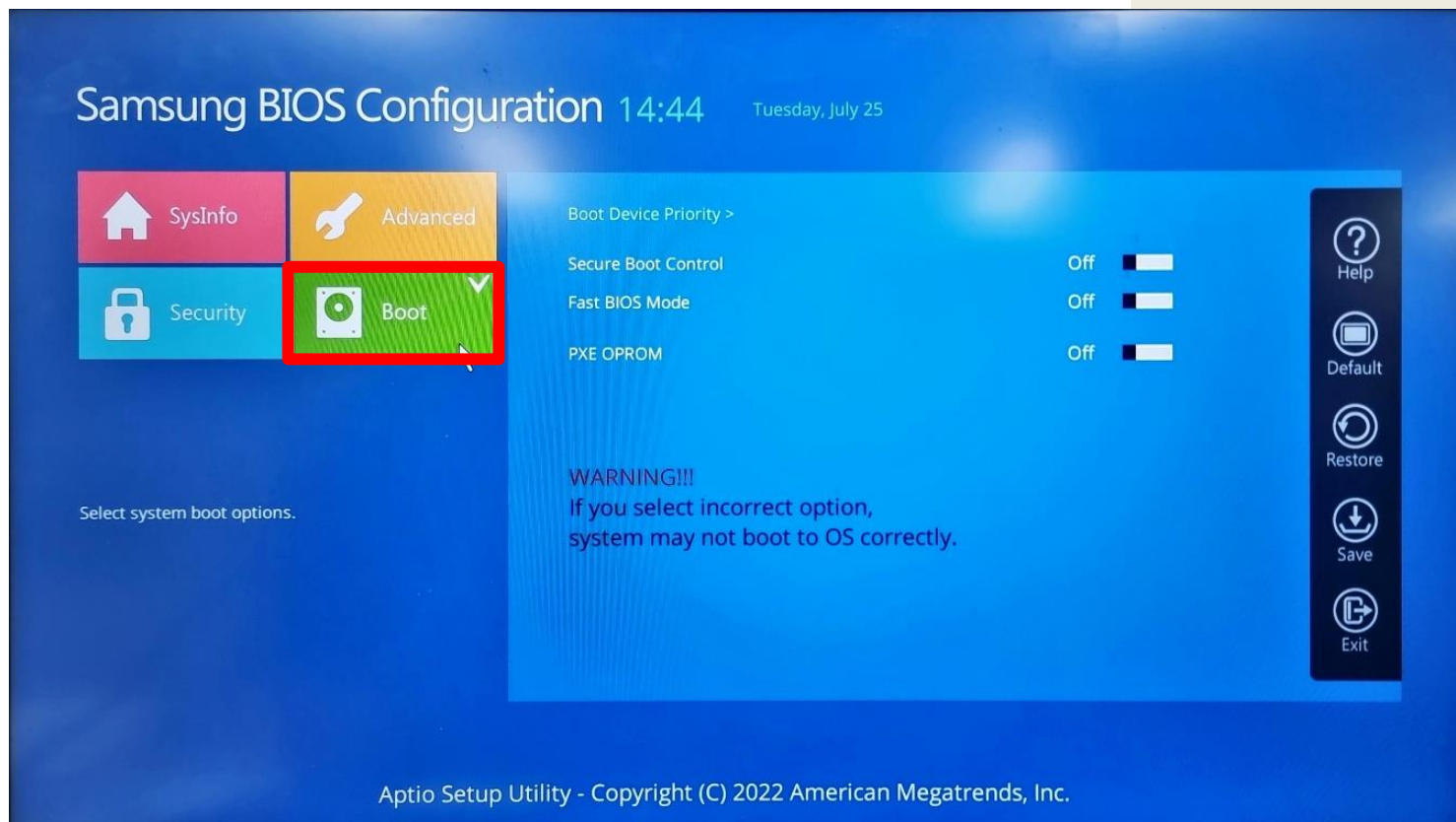
USB 장치로 부팅할 경우, BIOS 설정해야 함

- 전원을 켜 후에, '**F2**' 버튼을 누름(브랜드별로 키, 화면 다름)
- 화살표를 이용하여 '**Boot**' 메뉴로 이동함

(예시) 삼성 브랜드

[브랜드별 BIOS 모드 진입 방법]

- Acer: F2 또는 DEL 키
- ASUS: F2 (모든 PC), F2 또는 DEL 키 (메인보드)
- Dell: F2 또는 F12
- HP: ESC 또는 F10
- Lenovo: F2 또는 Fn + F2
- Lenovo (Desktops): F1
- Lenovo (ThinkPads): Enter + F1.
- MSI: DEL 키(메인보드 및 PC)
- Microsoft Surface 태블릿: 볼륨 증가 버튼을 길게 누름
- Origin PC: F2
- Samsung: F2
- Sony: F1, F2, F3 중 하나
- Toshiba: F2



BIOS 설정-USB 부팅

- 상단 '**Boot Device Priority**'를 클릭
- Boot Option #1을 '**UEFI:USB**'로 선택
- '**Save**'버튼 클릭
- setup confirmation 창에 '**ok**' 클릭

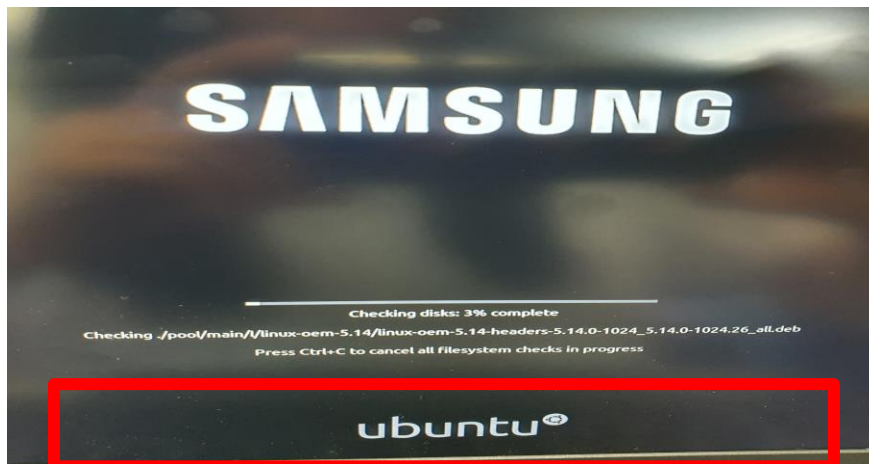
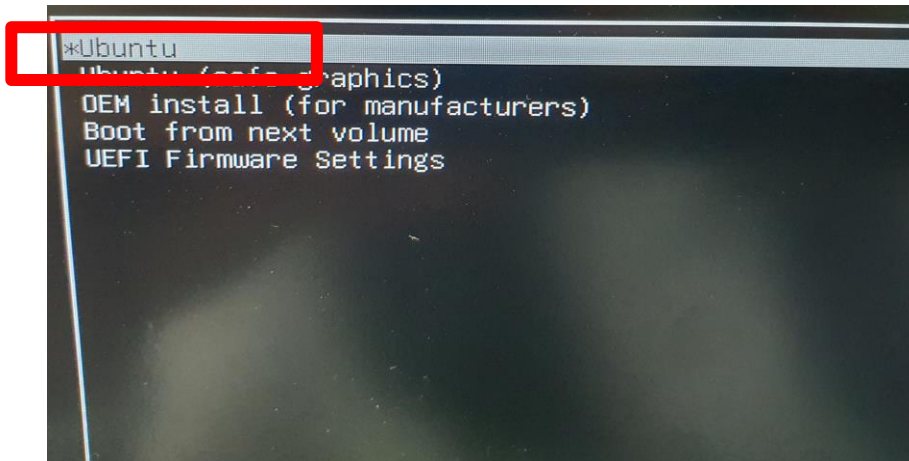
설치된 OS 개수에 따라 Boot Option#이 나열됨
Boot Option #1~#n 순서로 부팅함
제일 우선순위가 높은 #1의 OS에 'USB'를 설정해야함.
만약 'UEFI:USB'가 보이지 않는다면 부팅디스크를 만드는 시점에서 에러가 난 것이므로 다시 생성



운영체제 선택-Ubuntu

여러 운영체제가 설치된 경우 아래 화면이 보임

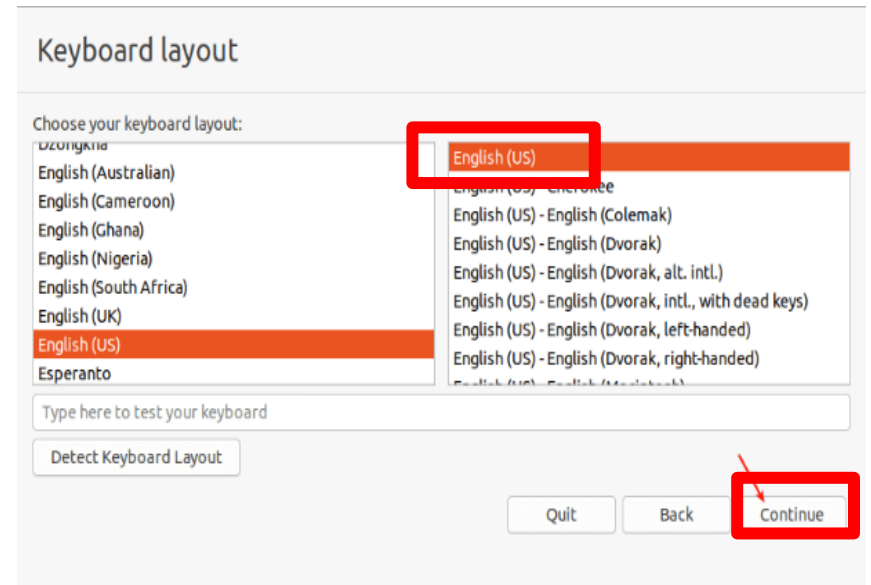
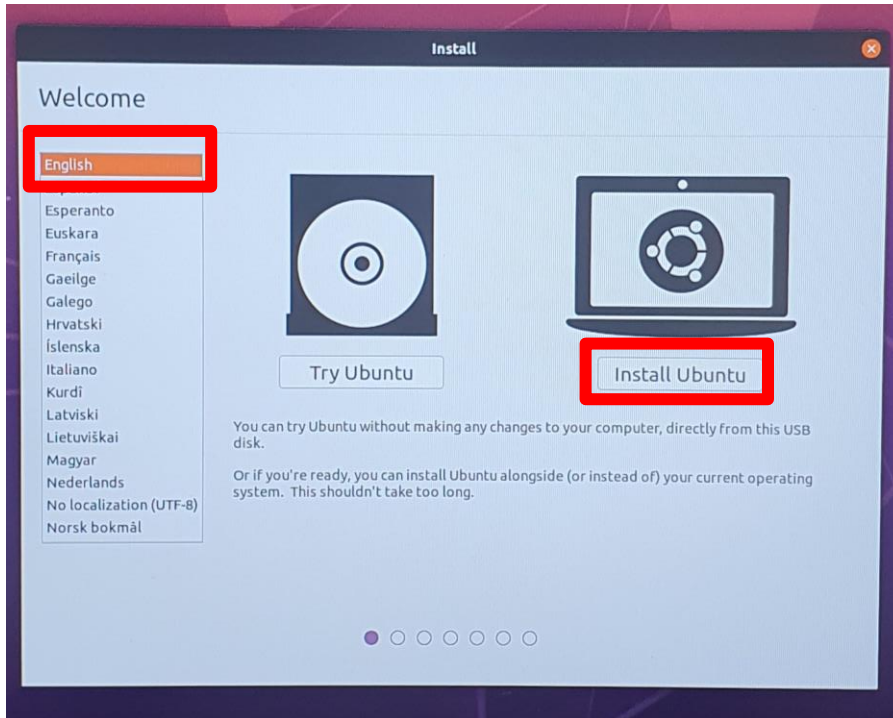
- 'Ubuntu' 클릭



화면 하단에 'ubuntu'라는 글자가 등장하며 설치가 진행되는 것임

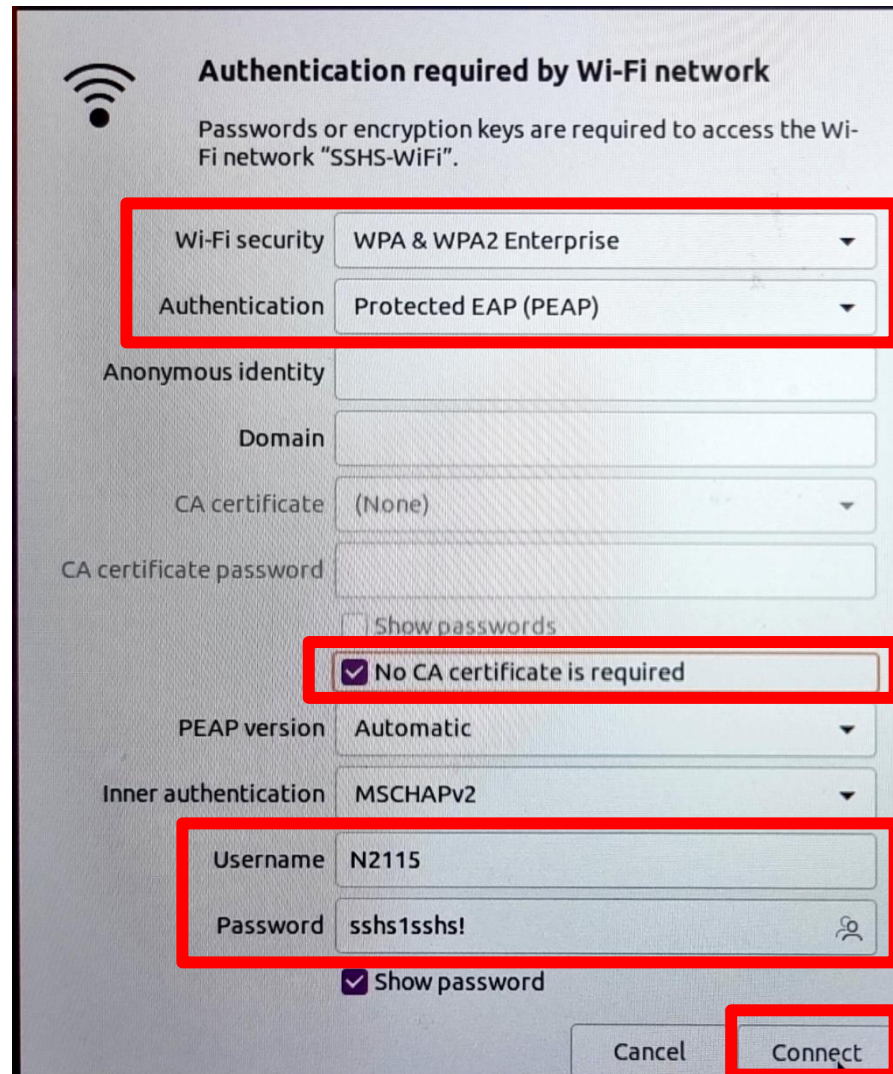
언어 설정-English

- 언어는 '**English**' 선택 (폴더명이 영어로 구성되어야 하므로 한글 선택 안함).
- '**Install Ubuntu**' 선택
- Keyboard layout은 '**English(US)**' 클릭 후 '**Continue**' 클릭



Wi-Fi 설정

- Wi-Fi security는 '**WPA & WPA2 Enterprise**' 선택
- Username, Password 선택
- '**Connect**' 클릭



The image shows a Wi-Fi authentication configuration window titled "Authentication required by Wi-Fi network". It contains several fields and options for setting up a secure connection to a network named "SSHS-WiFi".

Fields and options shown:

- Wi-Fi security:** WPA & WPA2 Enterprise
- Authentication:** Protected EAP (PEAP)
- Anonymous identity:** (empty field)
- Domain:** (empty field)
- CA certificate:** (None)
- CA certificate password:** (empty field)
- ☐ Show passwords
- ☒ No CA certificate is required
- PEAP version:** Automatic
- Inner authentication:** MSCHAPv2
- Username:** N2115
- Password:** sshs1sshs!
- ☒ Show password
- Buttons:** Cancel, Connect

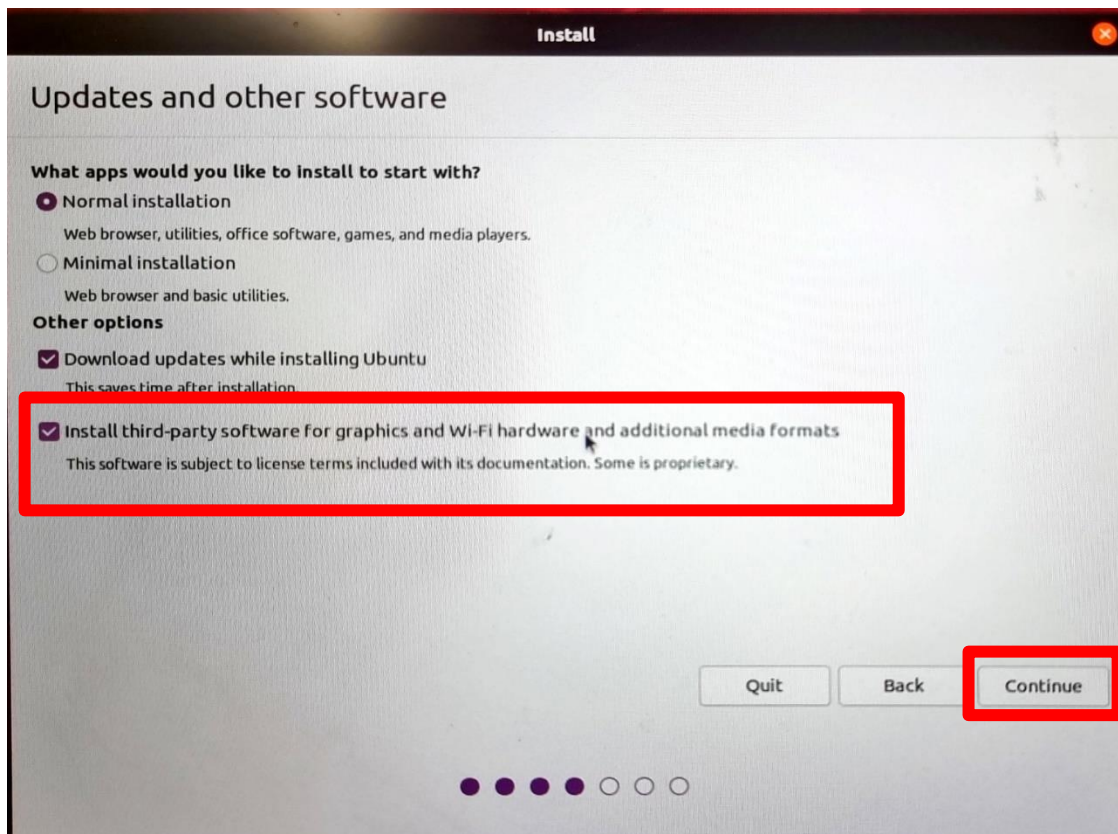
초기 환경 설정

- 마지막 체크 후 '**Continue**' 클릭

일반 설치

추가 미디어 포맷 지원을 내려받아 설치

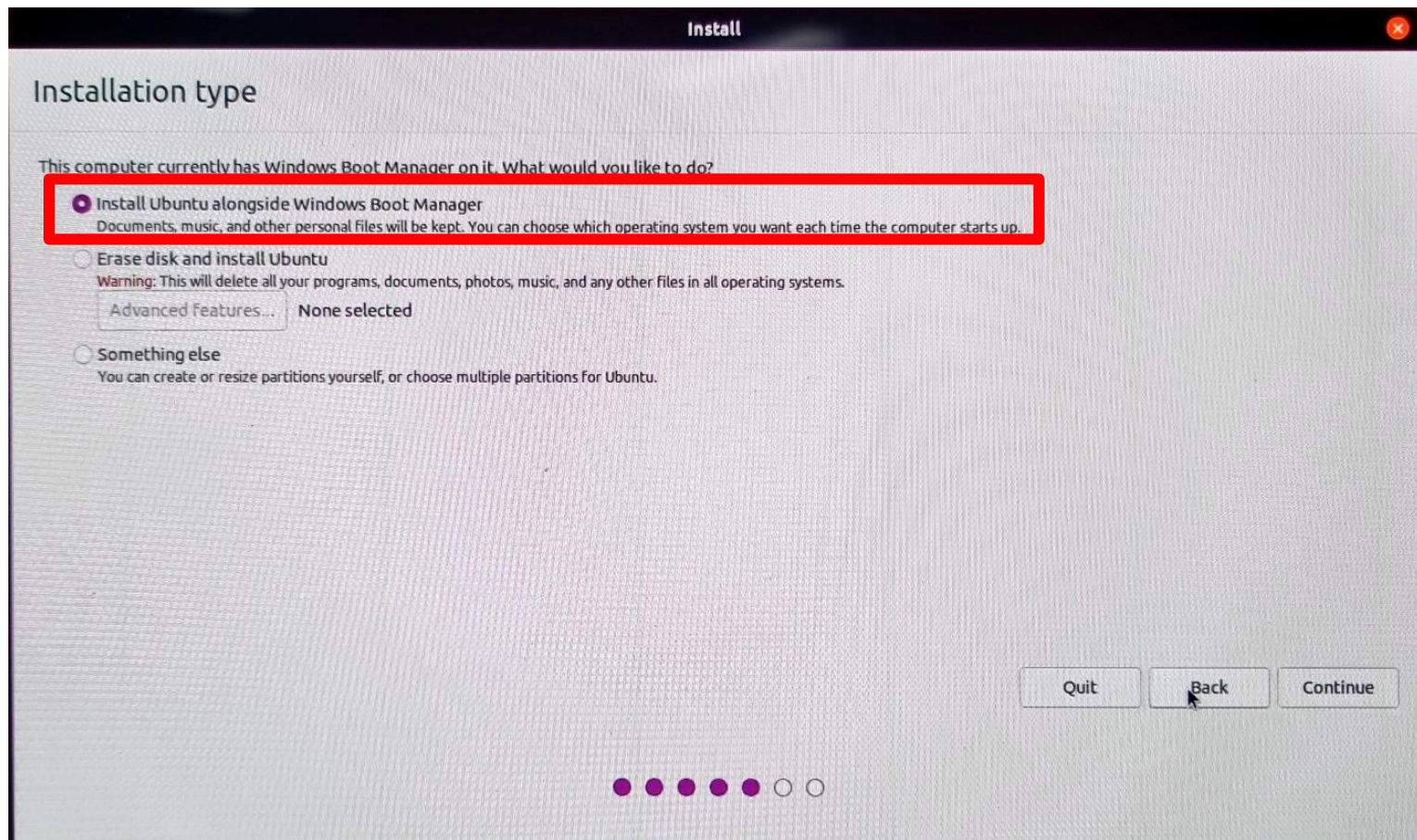
그래픽과 Wi-Fi 하드웨어를 위한 서드파티 소프트웨어 설치



초기 환경 설정

- 첫번째 체크 후 '**Continue**' 클릭

컴퓨터를 시작할 때 마다 원하는 운영체제를 선택함

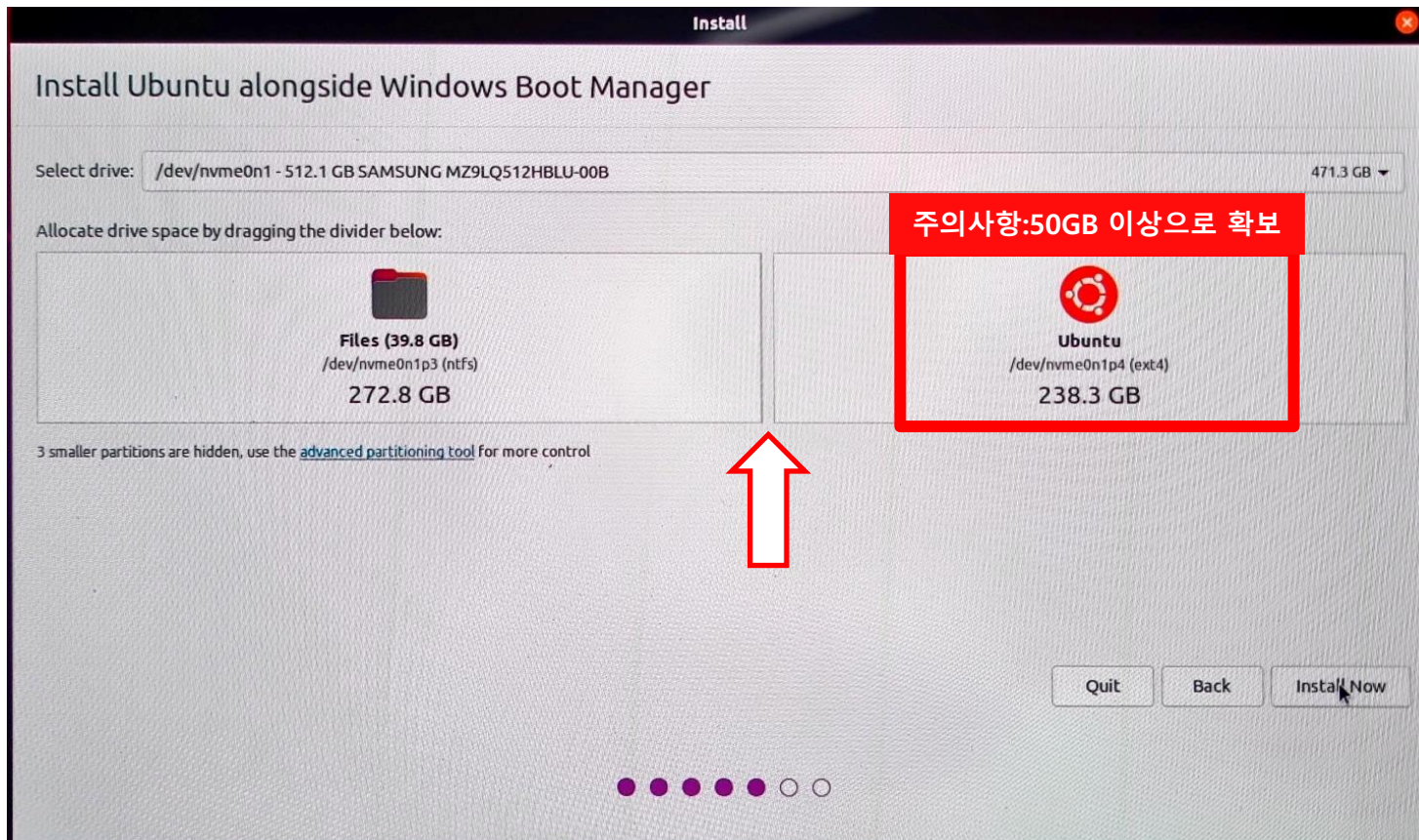


파티션

- 파티션 크기를 나누기

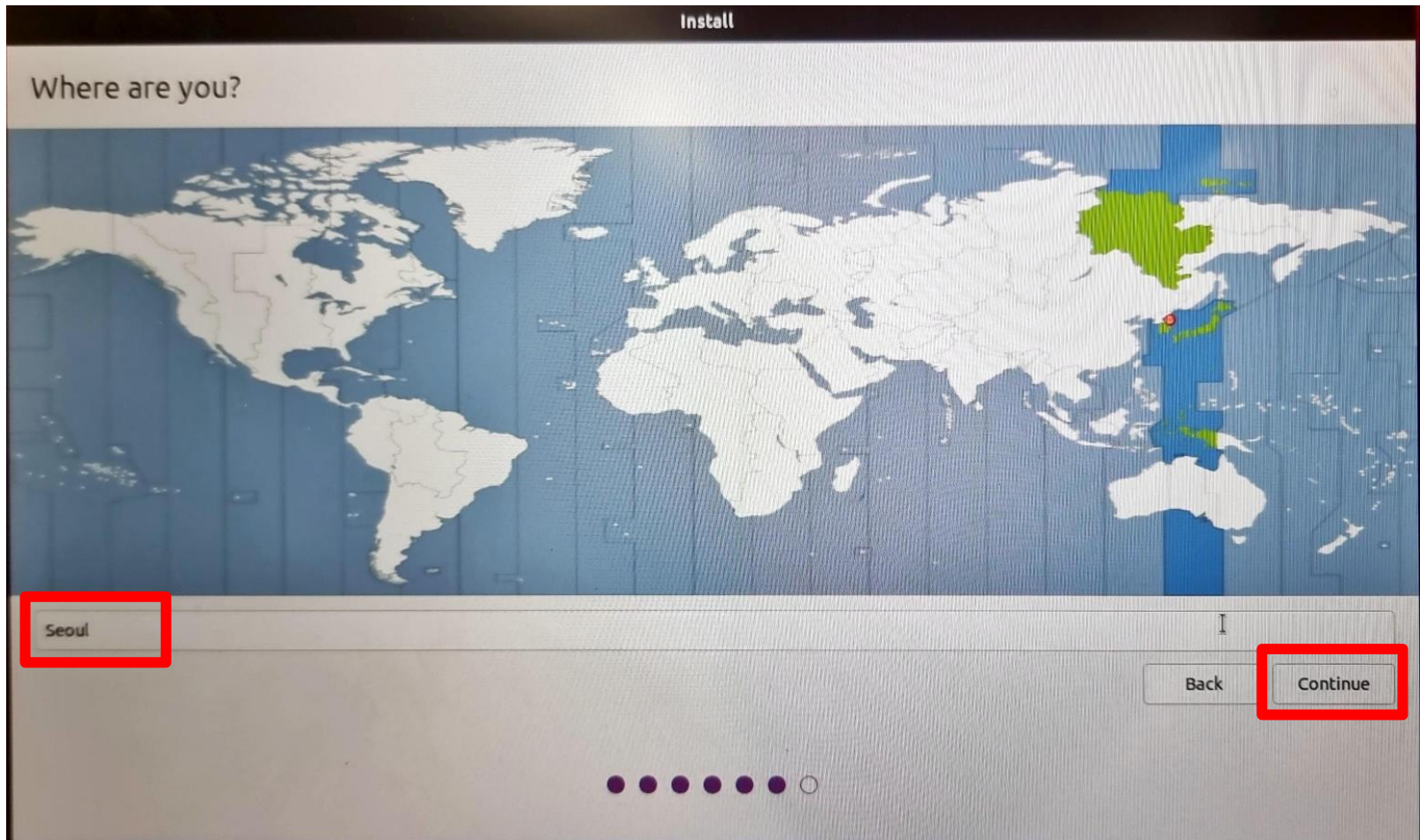
자동으로 메모리 분할이 이루어짐, 우분투 50GB 이상 확보 필요함

사이 영역(화살 표시)을 드래그 하여 메모리 조절 가능



나라

- 나라 'Seoul' 선택 후 'Continue' 클릭



Ubuntu 계정

- Pick a username은 성명이니셜+학번으로 입력 예) '**ksh3717**'
- password는 '**sshs1sshs!**'로 통일해서 입력
- '**Continue**' 클릭

The screenshot shows the 'Who are you?' screen during Ubuntu installation. The fields are as follows:

- Your name: TecMint How-Tos (with a green checkmark)
- Your computer's name: ubuntu-20.04 (with a green checkmark)
- Pick a username: tecmint (with a green checkmark)
- Choose a password: [masked] (with a green checkmark and 'Good password' text)
- Confirm your password: [masked] (with a green checkmark)

At the bottom, there are two radio buttons: 'Log in automatically' (unselected) and 'Require my password to log in' (selected). The 'Continue' button is highlighted with a red box.

✓ **Root계정 꼭 기억하기**

✓ **성명이니셜+학번**

✓ ex. 이름이 '김소희'이고, 학번이 '3717'인 경우 → **ksh3717**

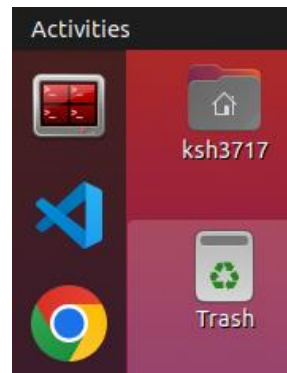
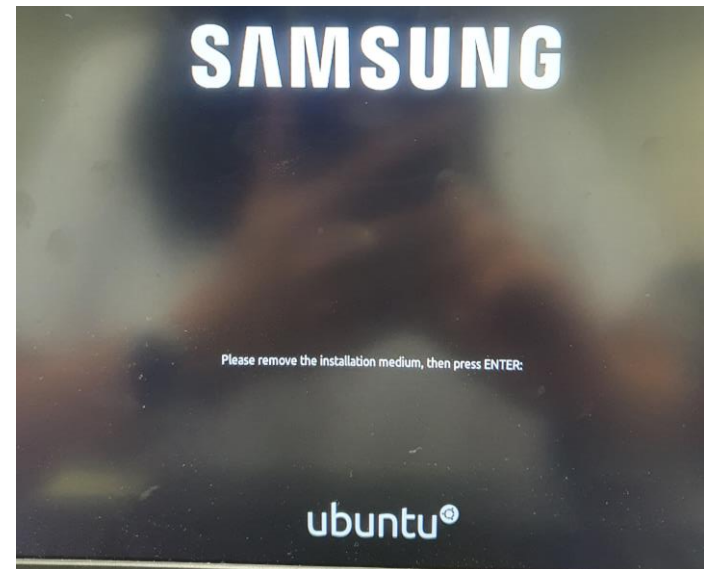
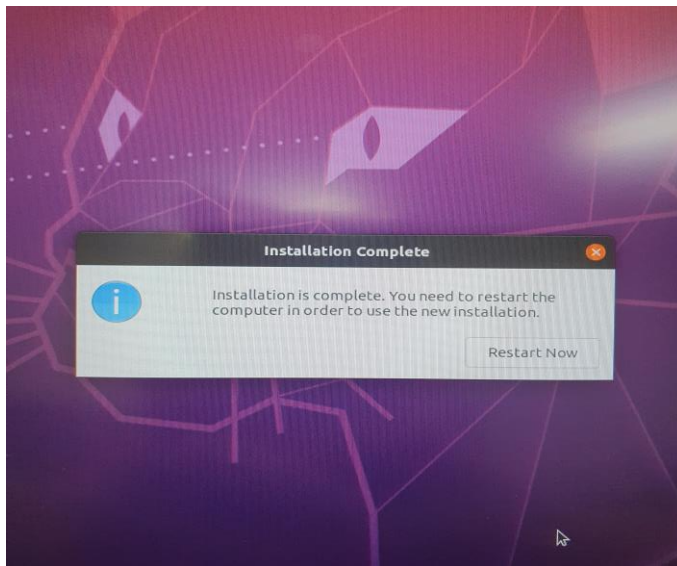
✓ 비번은 **sshs1sshs!** 로 통일

Ubuntu 설치 완료

기타 설치를 모두 마친 다음 'Installation complete'창이 뜸

재부팅 과정에서 'Please remove the installation medium, then press ENTER' 문구가 등장

USB 제거 후 ENTER키 누름



[설치를 끝낸 화면]

사용자 이름과 동일한 이름의 아이콘이 보임

22.04로 업그레이드하는 팝업이 보이면 Cancel을 선택

프로그램 설치 전

▪ **sudo apt update**

- 업데이트할 패키지들의 목록을 최신으로 갱신한다. 실제로 패키지를 업그레이드 하는게 아니다.
- E:~ 오류시: `sudo apt-get update`

▪ **sudo apt upgrade**

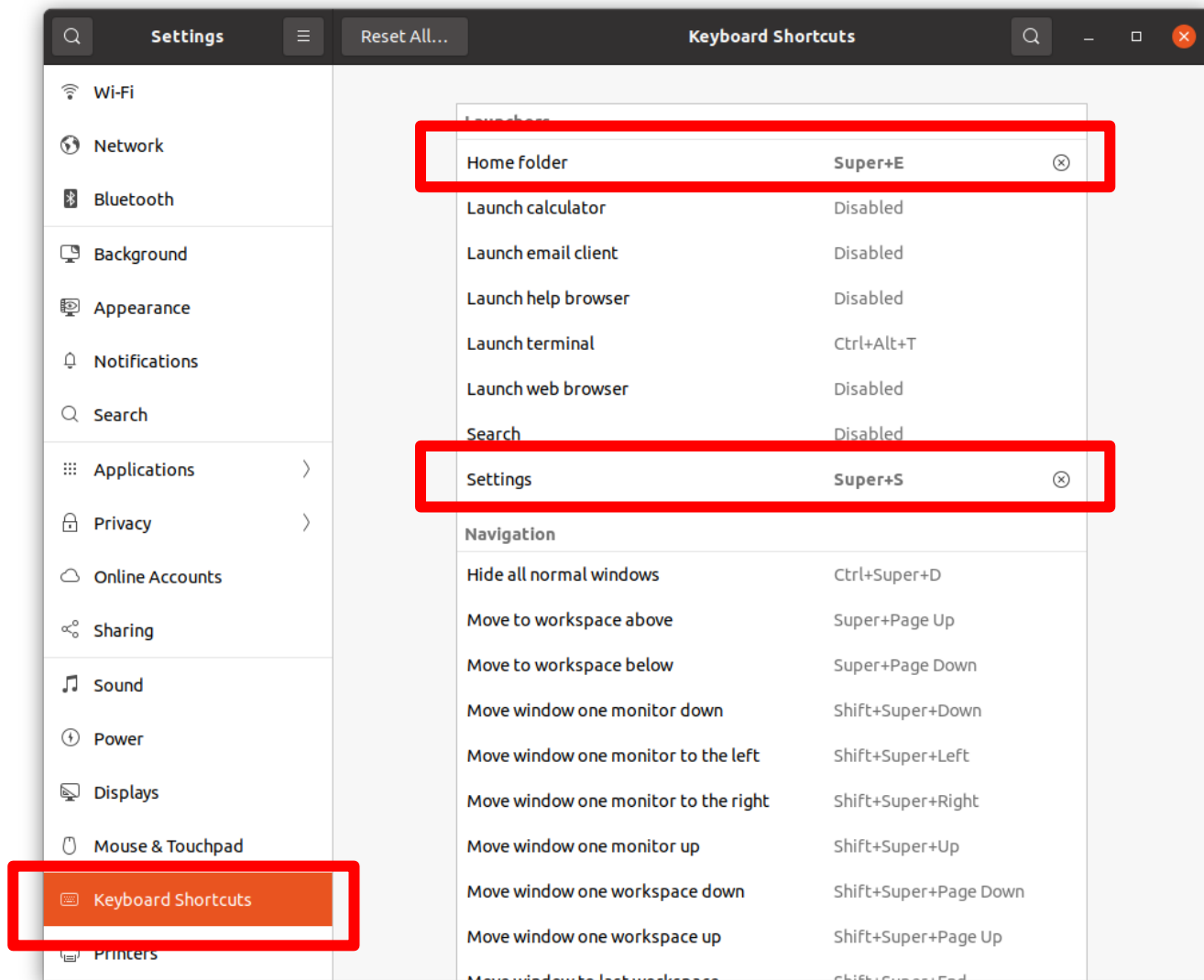
- 모든 패키지를 최신 버전으로 업그레이드한다.
- update 로 갱신된 최신 패키지들을 업그레이드 한다.

추가 프로그램 설치 및 설정 변경

Ubuntu	설치 및 설정 변경	실행
설치 환경	sudo apt update sudo apt upgrade	
Terminator	sudo apt install terminator	단축키 ctrl+alt+t
Terminator 설정	터미널에서 오른쪽 마우스 클릭 Preferences 클릭 Keybindings 탭 Copy의 Keybinding에서 더블클릭해서 Ctrl+C Paste의 Keybinding에서 더블클릭해서 Ctrl+V	터미널 창 ctrl+shift+t 수직 분할 ctrl+shift+e 수평 분할 ctrl+shift+o 창 제거 ctrl+shift+w
Visual Studio Code	https://code.visualstudio.com/ 접속 .deb 파일 다운로드 cd ~/Downloads (다운로드 폴더로 이동) sudo apt install ./파일명 (설치 실행)	터미널 code
Chrome	https://google.com/chrome/ 접속 .deb 파일 다운로드 cd ~/Downloads (다운로드 폴더로 이동) sudo apt install ./파일명 (설치 실행)	
영한키 설정	https://shanepark.tistory.com/231 참고	

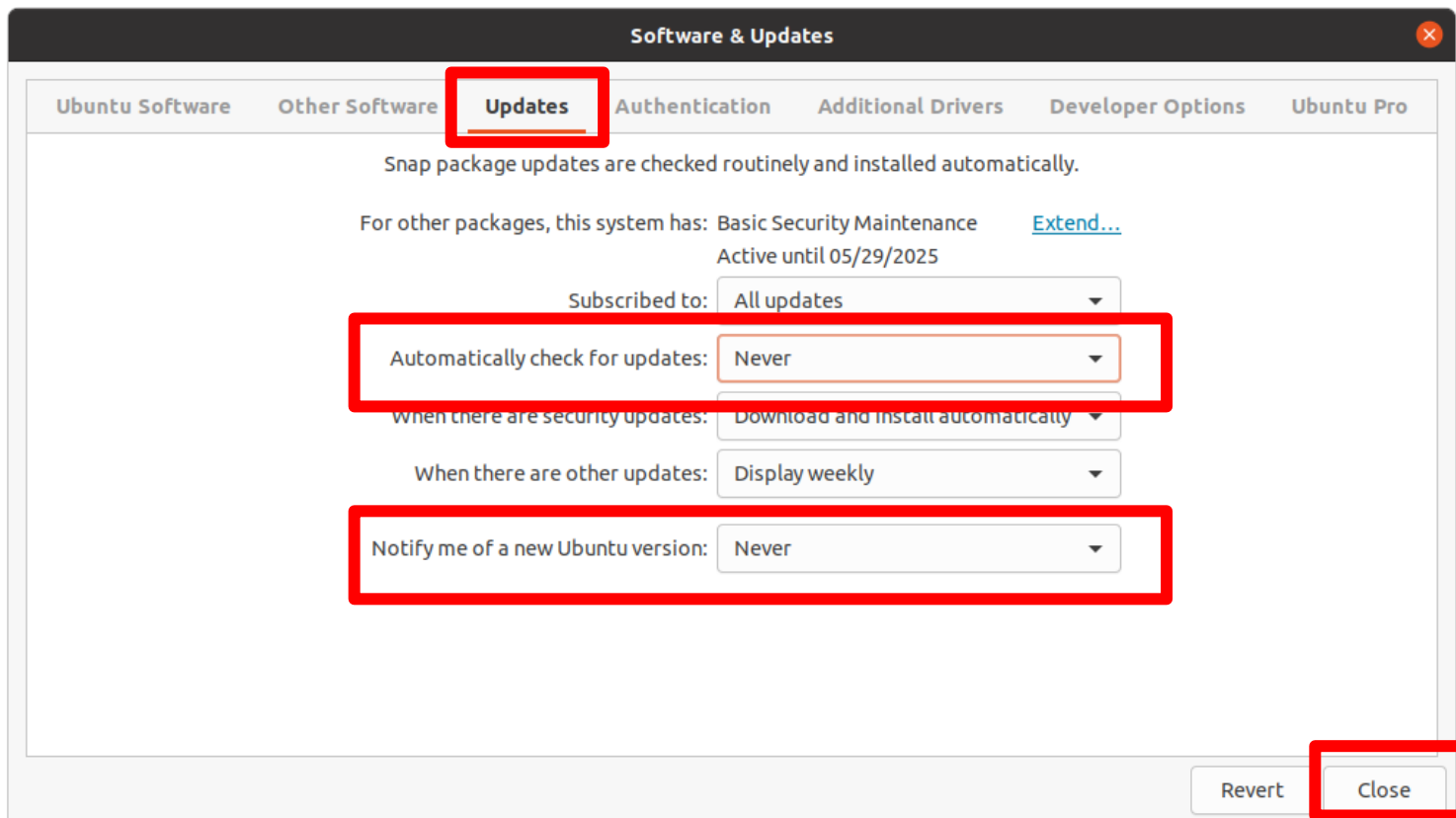
키보드 단축키 설정

- Settings - Keyboard Shortcuts



Ubuntu 버전 업데이트 알림 해제

- 왼쪽 아래  클릭
- Type to Search에 '**Software & Updates**' 입력 후 선택
- '**Updates**' 탭 선택
- 맨 아래 Notify를 '**Never**'로 변경 후 '**Close**' 클릭





ROS 2 foxy

PC 설치



설치 참고 사이트

- ROS 2 foxy 설치

- <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/#pc-setup>
- 링크 접속 후 버전인 **Foxy**를 반드시 선택

The screenshot shows the TurtleBot3 documentation website. At the top, there is a search bar with the text "Enter Search Terms" and a magnifying glass icon. Below the search bar, a navigation menu lists several versions: Kinetic, Melodic, Noetic, Dashing, **Foxy** (highlighted with a red box), Humble, and Windows. On the left side, a sidebar menu lists the following items: TurtleBot3, 1. Overview, 2. Features, 3. Quick Start Guide, **3. 1. PC Setup** (highlighted with a red box), 3. 2. SBC Setup, and 3. 3. OpenCR Setup. The main content area displays the heading "3. 1. PC Setup" followed by a "WARNING" box stating: "The contents in this chapter corresponds to the Remote PC (your desktop computer) TurtleBot3." Below this is another "Compatibility WARNING" box stating: "Jetson Nano does not support Ubuntu 20.04 and later. Please refer to NVIDIA developer page." The "Remote PC" and "Jetson Nano" terms are enclosed in rounded rectangular boxes.

ROS 2 foxy PC 설치

- 터미널(Ctrl+Alt+T)에서 다음 코드 실행
 - ros2 foxy 설정이 다 되어 있는 ROBOTIS파일 설치
 - Ubuntu에서 ROS2 foxy 설치하는 다음 guide 확인
 - <https://docs.ros.org/en/foxy/Installation/Ubuntu-Install-Debians.html>

```
$ sudo apt update
```

```
$ sudo apt upgrade
```

```
$ wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis_tools/master/install_ros2_foxy.sh
```

```
$ sudo chmod 755 ./install_ros2_foxy.sh
```

```
$ bash ./install_ros2_foxy.sh
```

ROS2 추가 프로그램 설치

- Gazebo11 - 3D 시뮬레이션 환경

```
$ sudo apt-get install ros-foxy-gazebo*
```

```
$ sudo apt-get install ros-foxy-turtlebot3-gazebo
```

- Cartographer - SLAM(동시적 위치 추정 및 지도 작성)

```
$ sudo apt install ros-foxy-cartographer
```

```
$ sudo apt install ros-foxy-cartographer-ros
```

- Navigation2 - 길찾기

```
$ sudo apt install ros-foxy-navigation2
```

```
$ sudo apt install ros-foxy-nav2-bringup
```

TurtleBot3 설치

- TurtleBot3 via Debian Packages.
 - ROS2 교육용 로봇인 TurtleBot3 설치
 - TurtleBot3는 메타패키지(다양한 패키지의 묶음)임.

```
$ source ~/.bashrc
```

```
$ sudo apt install ros-foxy-dynamixel-sdk
```

```
$ sudo apt install ros-foxy-turtlebot3-msgs
```

```
$ sudo apt install ros-foxy-turtlebot3
```

도메인 아이디 설정

- Remote PC에서 ROS 환경 설정
 - 여러명이 한 공간에서 ROS2를 사용할 경우 시스템 충돌을 방지하기 위해 도메인 아이디를 다르게 설정할 수 있음
 - ROS_DOMAIN_ID 범위는 0~232
 - Remote PC와 Robot의 도메인 아이디는 같아야 함

```
$ echo 'export ROS_DOMAIN_ID=30 #TURTLEBOT3' >>  
~/.bashrc  
$ source ~/.bashrc
```

환경 변수 확인

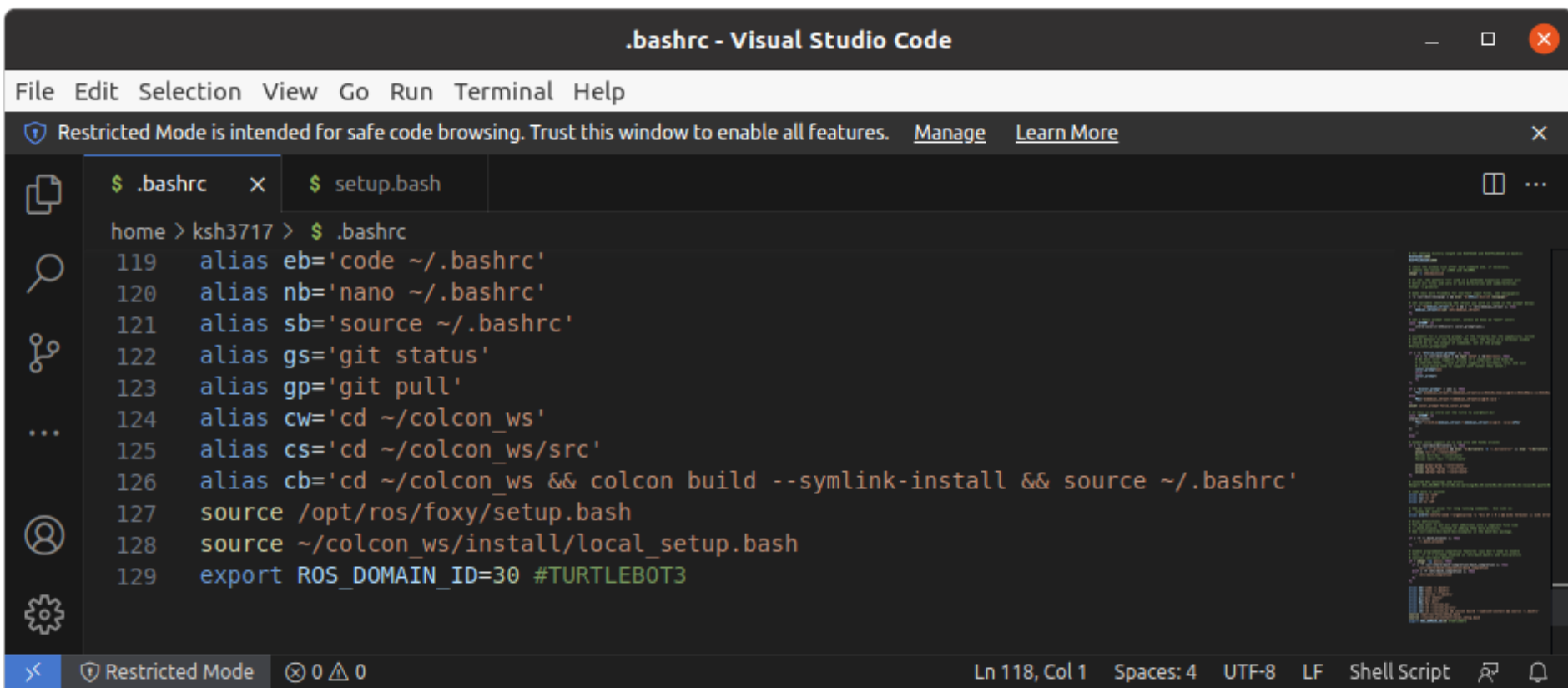
- 환경 변수
 - 배치 파일을 source 해주면 환경 변수가 시스템에 등록됨
- 환경 변수 확인
 - 명령어 실행 후 다음 내용 확인하기
 - ROS_VERSION=2
 - ROS_PYTHON_VERSION=3
 - ROS_DISTRO=foxy
 - ROS_DOMAIN_ID=30 (설정한 숫자)

```
$ printenv | grep -i ROS
```

배치파일(~/.bashrc)

■ 시스템 배치 파일

- 배치파일 중 '~/.bashrc' 파일은 터미널이 열릴 때마다 적용됨
 - 다른 배치파일은 'source 배치파일' 명령어를 실행해야 적용됨
- 주의 colcon_ws는 사용자가 생성한 폴더로 변경 가능함



```
.bashrc - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

$ .bashrc x $ setup.bash
home > ksh3717 > $ .bashrc
119 alias eb='code ~/.bashrc'
120 alias nb='nano ~/.bashrc'
121 alias sb='source ~/.bashrc'
122 alias gs='git status'
123 alias gp='git pull'
124 alias cw='cd ~/colcon_ws'
125 alias cs='cd ~/colcon_ws/src'
126 alias cb='cd ~/colcon_ws && colcon build --symlink-install && source ~/.bashrc'
127 source /opt/ros/foxy/setup.bash
128 source ~/colcon_ws/install/local_setup.bash
129 export ROS_DOMAIN_ID=30 #TURTLEBOT3

Ln 118, Col 1 Spaces: 4 UTF-8 LF Shell Script
```

배치파일 수정 및 적용

- 배치파일 확인

```
$ code ~/.bashrc
```

- 배치파일 수정

- 열린 배치파일에서 다음 내용을 추가
- 주의) 도메인 숫자 변경하기, 작성시 '=' 전후 띄어쓰기하지 않기
- 주의) LDS_MODEL은 라이더 모델을 확인하기

```
export ROS_DOMAIN_ID=30
export TURTLEBOT3_MODEL=waffle_pi
export LDS_MODEL=LDS-02
```

- 배치파일 적용

```
$ source ~/.bashrc
```


배치파일 수정 및 적용

- 배치파일 확인

```
$ code ~/.bashrc
```

- 배치파일 수정

- 열린 배치파일에서 다음 내용을 추가함

```
alias eb='code ~/.bashrc'
```

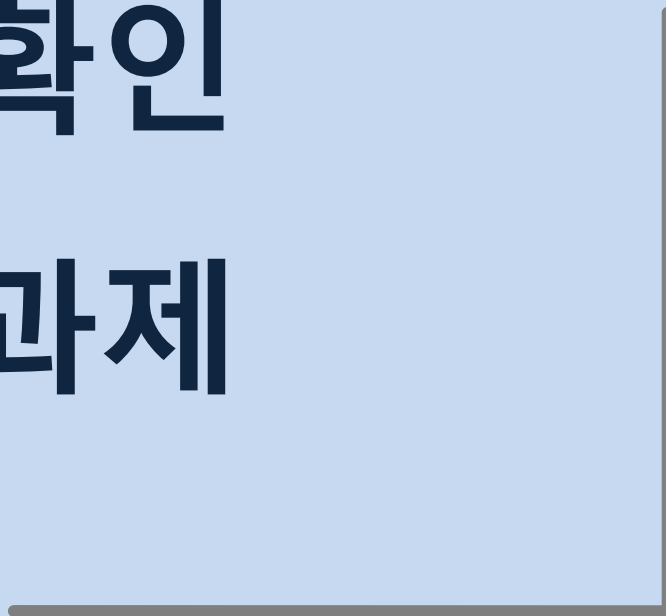
- 배치파일 적용

- 수정된 내용을 적용하려면 다음 명령어를 실행해야 함

```
$ source ~/.bashrc
```



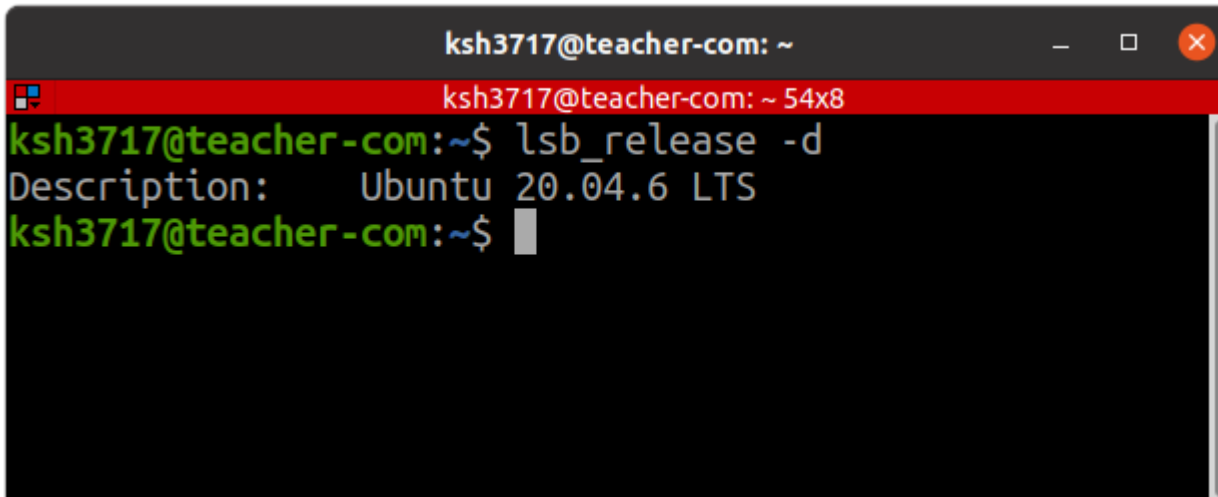
설치 확인 실습 과제



터미네이터 창 열어 계정 규칙 확인

과제1

- terminator 창을 열었을 때 다음과 같이 '사용자 계정'과 '우분투 버전'이 보이도록 화면 캡처하기
- 사용자 계정 규칙
 - '**성명이니셜+학번**'. 녹색 텍스트 @앞은 사용자 계정을 의미함
- 우분투 버전 표시
 - 터미널에서 '**lsb_release -d**' 명령어 입력



```
ksh3717@teacher-com: ~  
ksh3717@teacher-com: ~ 54x8  
ksh3717@teacher-com:~$ lsb_release -d  
Description:    Ubuntu 20.04.6 LTS  
ksh3717@teacher-com:~$
```

Publisher – Subscriber 노드 구성

- ROS는 프로그램의 재사용성을 극대화하기 위해 최소 단위의 실행 가능한 프로세스인 노드(Node) 단위로 프로그램을 작성함
- 하나 이상의 노드 또는 노드 실행을 위한 정보 등을 묶어 놓은 것을 패키지(Package)라고 함
- 패키지의 묶음을 메타패키지(Metapackage)라고 함

터미널을 2개 열고 아래 코드를 각각 실행시켜 보자

```
$ ros2 run demo_nodes_cpp listener
```

```
$ ros2 run demo_nodes_cpp talker
```

실행결과가 나온다는 것은 이미 지정된 DDS(RMW)를 사용하고 있는 것임

```
$ rqt_graph
```

실행된 두 노드 간 토픽을 주고 받고 있다는 것을 알 수 있음

동그라미는 노드, 네모는 토픽 또는 액션, 화살표는 메시지의 방향을 의미
서비스는 필요할 때 순간적으로 사용되는 형식이라 표시되지 않음

Publisher – Subscriber 노드 구성

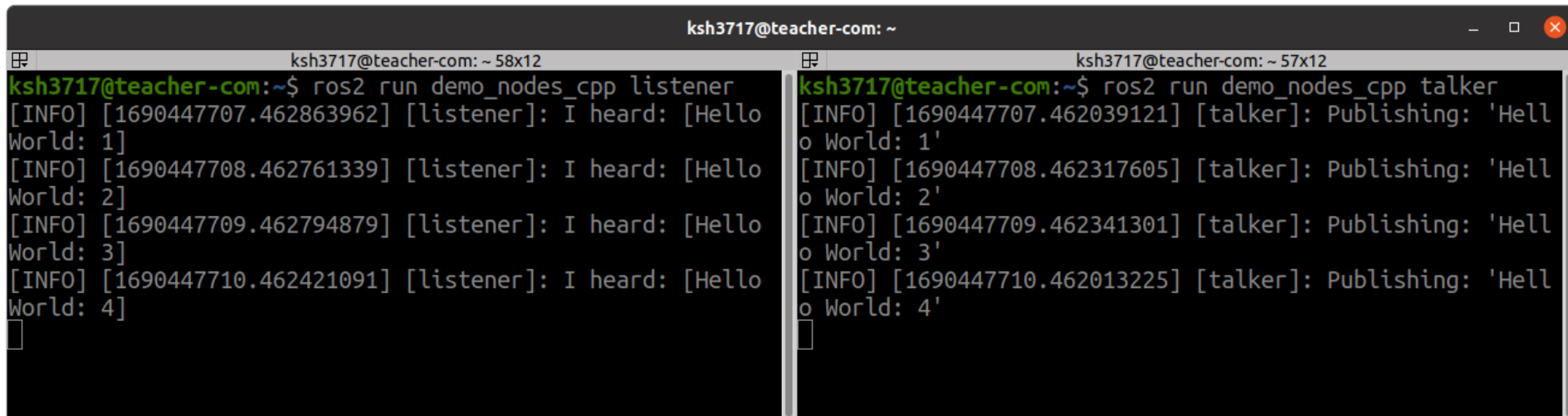
과제2

터미널을 2개 열고 아래 코드를 각각 실행시킨 다음 캡처하기

```
$ ros2 run demo_nodes_cpp listener
```

```
$ ros2 run demo_nodes_cpp talker
```

- 터미널 창을 분할함
- listener, talker를 각 창에 실행함



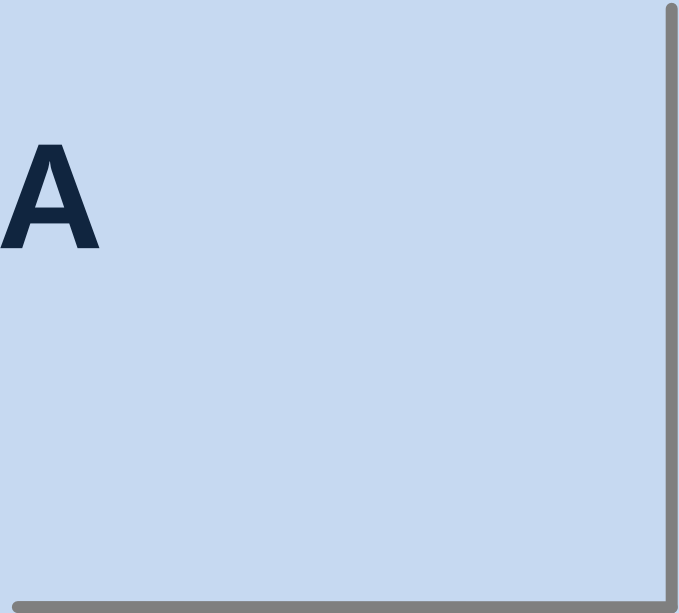
The image shows two terminal windows side-by-side. The left window is titled 'ksh3717@teacher-com: ~ 58x12' and shows the output of the 'listener' node. The right window is titled 'ksh3717@teacher-com: ~ 57x12' and shows the output of the 'talker' node. Both windows show a series of 'Hello World' messages being published and received.

```
ksh3717@teacher-com: ~ 58x12
ksh3717@teacher-com:~$ ros2 run demo_nodes_cpp listener
[INFO] [1690447707.462863962] [listener]: I heard: [Hello
World: 1]
[INFO] [1690447708.462761339] [listener]: I heard: [Hello
World: 2]
[INFO] [1690447709.462794879] [listener]: I heard: [Hello
World: 3]
[INFO] [1690447710.462421091] [listener]: I heard: [Hello
World: 4]
█

ksh3717@teacher-com: ~ 57x12
ksh3717@teacher-com:~$ ros2 run demo_nodes_cpp talker
[INFO] [1690447707.462039121] [talker]: Publishing: 'Hell
o World: 1'
[INFO] [1690447708.462317605] [talker]: Publishing: 'Hell
o World: 2'
[INFO] [1690447709.462341301] [talker]: Publishing: 'Hell
o World: 3'
[INFO] [1690447710.462013225] [talker]: Publishing: 'Hell
o World: 4'
█
```



Q&A



apt(Advanced Packaging Tool)

- 리눅스 시스템에 패키지를 설치하고 제거하는데 사용된다.

apt 명령	기존 명령	설명
apt install	apt-get install	패키지 목록
apt remove	apt-get remove	패키지 삭제
apt purge	apt-get purge	패키지와 관련 설정 제거
apt update	apt-get update	레파지토리 인덱스 갱신
apt upgrade	apt-get upgrade	업그레이드 가능한 모든 패키지 업그레이드
apt autoremove	apt-get autoremove	불필요한 패키지 제거
apt full-upgrade	apt-get dist-upgrade	의존성 고려한 패키지 업그레이드
apt search	apt-cache search	프로그램 검색
apt show	apt-cache show	패키지 상세 정보 출력

update, upgrade

▪ sudo apt update

- 업데이트할 패키지들의 목록을 최신으로 갱신한다. 실제로 패키지를 업그레이드 하는게 아니다.
- E:~ 오류시: sudo apt-get update

▪ sudo apt upgrade

- 모든 패키지를 최신 버전으로 업그레이드한다.
- update 로 갱신된 최신 패키지들을 업그레이드 한다.

▪ sudo apt dist-upgrade

- upgrade 는 의존성 체크를 하지 않고 업그레이드하지만
- dist-upgrade는 의존성 체크를 하면서 위의 upgrade에서 수행되지 못한 dependency 까지 설치해준다.