데이터사이언스를위한 지식관리시스템

Chap. 01

강의개요, 리눅스 서버 기초 - 1

본 과목의 목표

- 연구에 필요한 데이터를 잘 모으고 저장하는 스킬을 습득합니다
- 데이터 파이프라인의 흐름, ETL 방법론에 대해 집중합니다
- ETL: Extract, Transform, Load

데이터 파이프라인

- 데이터를 제공하는 곳은 많습니다
- 제때 수집하지 않으면 다시 수집하지 못할 수 있습니다
- 수집된 데이터를 제때 처리하지 않으면 데이터의 가치가 소멸할 수 있습니다
- 그래서 우리는 데이터 파이프라인 이라는 것을 만들고 운영합니다

우리가 사용할 도구

데이터가 잠시 대기하는 곳

이 도구들을 핸들링 할 언어

데이터의 출발

이 도구들이 설치될 플랫폼

데이터를 잘 저장하는 곳

각 도구의 역할

- Ubuntu
  - Kafka, MongoDB를 설치하고 python 언어로 이들을 핸들링 하는 기반 운영체제 입니다
  - Windows PC 를 사용할 수도 있지만 굳이 Ubuntu를 사용하는 이유는 잠시후에 알아봅니다

- Kafka
  - 하나의 원천 데이터만 수집한다면 사실 이 도구는 있을 필요성이 낮습니다
  - 그러나 데이터의 출처가 여러 곳이라면 한곳에 집중시켜서 스트리밍 하는 전략을 사용해볼 수 있습니다

- MariaDB, MongoDB, Neo4j
  - Kafka로부터 원천 데이터를 수집 저장합니다
  - 집계 결과를 저장하고 검색합니다
  - 관계형 데이터베이스, 문서지향 데이터베이스, 그래프형 데이터베이스로 구분

원천 데이터가 존재하는 곳

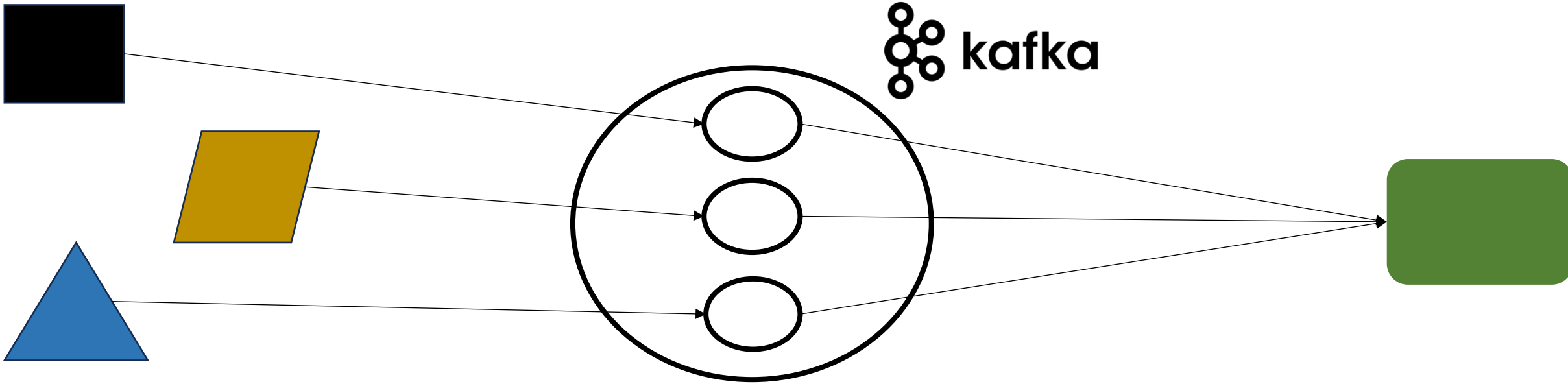
- 데이터를 제공해주는 분의 의지(?)에 달려있습니다
- 아래 각각의 경우에 우리는 데이터를 어떻게 가져올 수 있을지 생각해봅시다
- 데이터는 계속 쌓이고 있다고 가정합니다

➢ 파일: 파일서버에 접속해서 복사
➢ 데이터베이스: 주기적으로 쿼리하여 가져오기
➢ REST API: 주기적으로 호출해서 가져오기
➢ 기타 등등(?): IoT 디바이스, 웹사이트, 클라우드 서비스

✓ 결론: 데이터는 항상 우리가 원하는 방법으로 제공되지는 않는다

Kafka: 여러 Data source를 한곳에 집중하여 스트리밍

- 여러 Data source를 한곳에 퍼블리싱 합니다
- 우리는 이를 구독하면 여러 Data source를 하나의 프로토콜에 의해 순차적으로 스트리밍 받을 수 있습니다
- 결론적으로 다양한 Data source로부터 발생하는 복잡함과 무관하게 데이터를 어떻게 변환하고 저장할지에 집중할 수 있도록 합니다

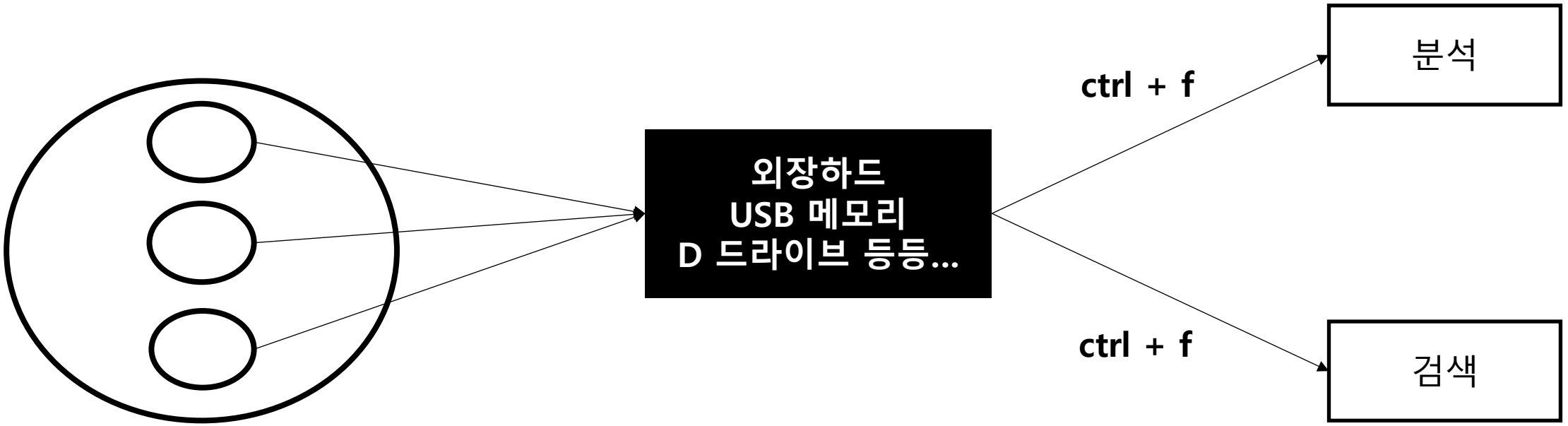


웹스크래핑

- 웹사이트로부터 데이터를 직접 수집하는 방법입니다
- 이 부분은 저작권 문제가 있을 수 있기 때문에 신중히 진행해야 합니다
- 웹스크래핑이 허용된 수준에 한해 진행합니다

수집한 데이터의 저장과 검색

- 그냥 파일로 저장하고 관리하는 것이 가능합니다
- 그러면 그 데이터를 파일 탐색기를 열고 각 파일로 저장, 검색하는 것을 생각해봅니다
- 그리고 파일 내에서 데이터를 찾기 위해 탐색하는 과정을 생각해봅니다



데이터베이스: 효과적인 분석, 검색환경을 제공

- 저장된 데이터의 Aggregation / CRUD 엔진을 내장
- 파일시스템 기반의 분석, 탐색보다 효과적으로 데이터를 관리할 수 있습니다

가장 먼저 필요한 것

- 리눅스 기초를 먼저 다루도록 하겠습니다
- 리눅스는 운영체제의 하나 입니다
- 운영체제란 여러가지 프로그램을 구동할 수 있는 플랫폼이라고 할 수 있습니다
- 운영체제가 없는 상황을 생각해보고 운영체제를 사용하는 이유를 생각해봅니다
- 사용하는 운영체제에 따라 발생하는 차이에 대해 알아봅니다
- 우리는 Ubuntu x64 를 사용하는 것으로 결론을 내리고 이 환경을 구성하겠습니다

운영체제를 사용할 때와 그렇지 않을 때

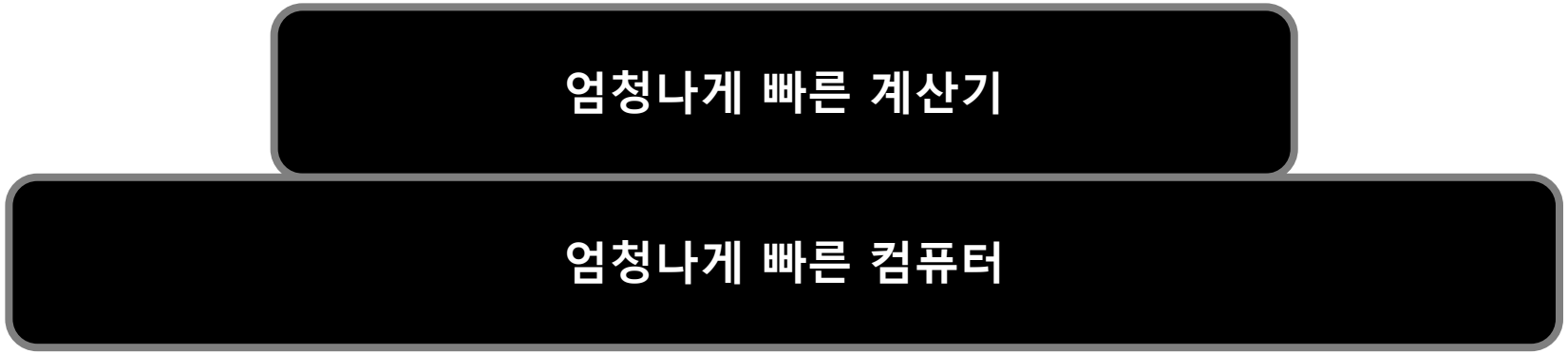
- 계산기를 운영체제가 있을 때와 없을 때로 생각해보겠습니다
- 화면 UI는 숫자만 표시하는 7-segment로 생각해보겠습니다

- 운영체제 없는 계산기
  - 화면 출력을 직접 구현
  - 0부터 1, A부터 Z를 어떻게 표시할지 직접 구현
  - 구현에 대한 생산성이 낮습니다

- 운영체제가 있는 계산기
  - 운영체제가 제공하는 함수를 이용해 0부터 1, A부터 Z를 표시
  - 이를 System call 이라고 합니다
  - 운영체제 위에서 구동되는 프로그램은 이 System call에 의존성을 갖습니다

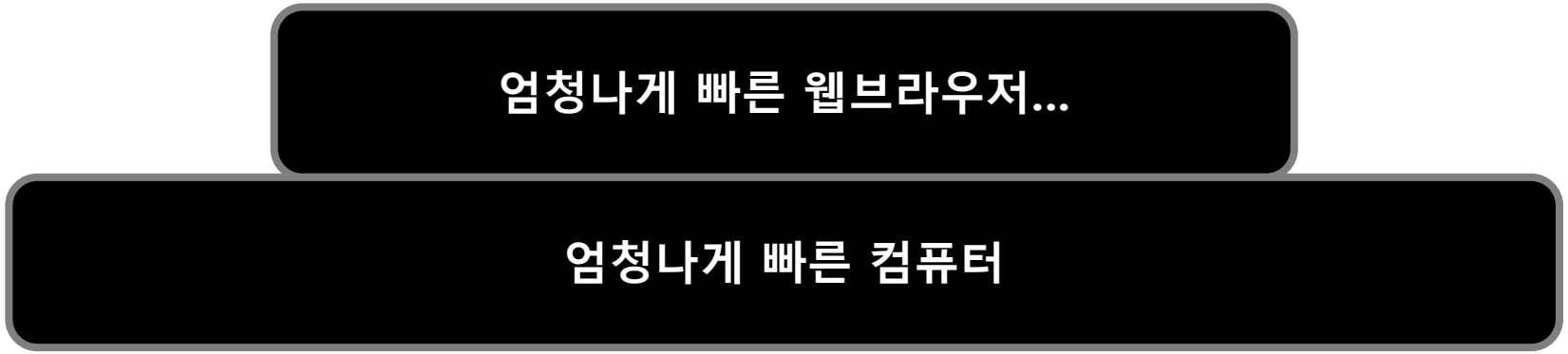
운영체제 없이 구동하는 컴퓨터

- 컴퓨터와 계산기 사이에 가로막고 있는 것이 아무것도 없습니다
- 생산성을 고려하지 않을 때, 구현에 드는 시간이 무한대라고 가정할 때 최고의 계산기를 만들 수 있겠습니다



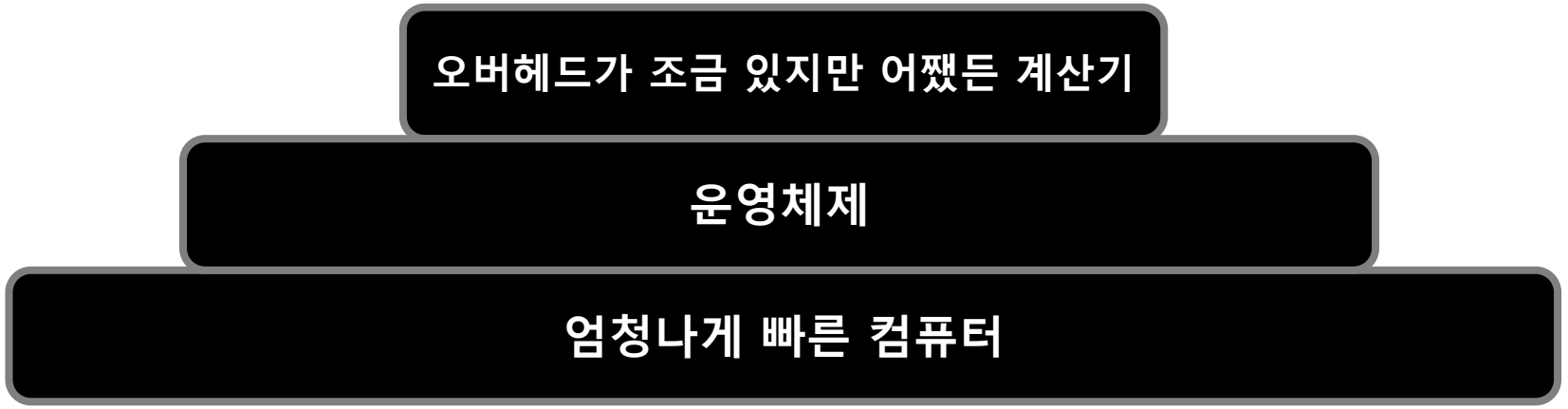
운영체제 없이 구동하는 컴퓨터

- 이제 이 컴퓨터 위에 웹브라우저를 구동한다고 생각해보겠습니다
- 해야할 일이 너무 많아집니다
- 화면 그리기부터 어떻게 해야할지 막막해집니다



운영체제 없이 구동하는 컴퓨터

- 이 경우 계산기는 컴퓨터를 제어하기 위해서 운영체제와 소통해야 합니다
- 따라서 운영체제의 포맷으로 소통해야 합니다(System call)
- 이를 오버헤드라고 표현하겠습니다
- (생산성을 고려하지 않을 때) 컴퓨터를 직접 제어하는 것보다는 빠르지 않습니다



운영체제 없이 구동하는 컴퓨터

- 그러나 이 공통된 포맷으로 여러가지 프로그램을 운영체제 위에서 돌릴 수 있습니다
- 한편으로는 이 공통된 포맷이 아니면 해당 운영체제 위에서 돌릴 수 없습니다
- Windows application, macOS application이 구분되는 이유 입니다

운영체제의 선택

- 서버용: 서버 용도로 최적화된 버전
- 데스크톱용: 데스크톱 용도로 최적화된 버전

- Windows, Windows Server
- Ubuntu Desktop, Ubuntu Server
- macOS, macOS Server

- 서버용 운영체제를 데스크톱용으로, 혹은 그 반대로의 사용은 가능합니다
- 그러나 일반적으로 그렇게 하지는 않습니다
- 그래서 서버용 운영체제를 선택합니다

운영체제의 선택

- Top server OS 2024, 혹은 그와 비슷한 키워드로 검색해봅시다
- 이 중에서 무료로 사용할 수 있는 것으로 범위를 줄여봅니다
- 그러면 대개는 아래와 같은 결론이 나옵니다

✓Ubuntu Server

✓데스크톱 사용자인 우리들에게는 낯설게 느낄 수는 있습니다

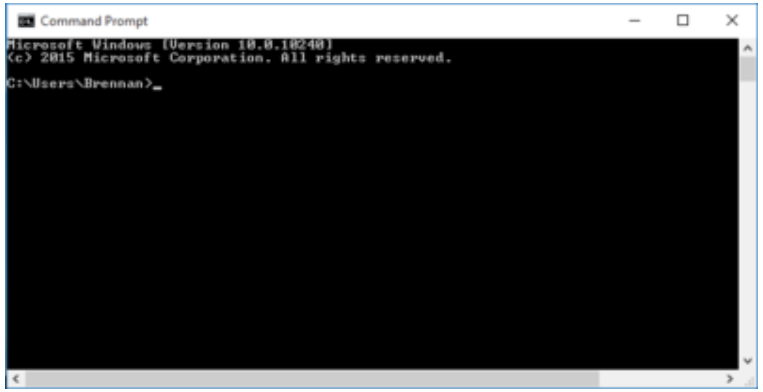
✓그러나 서버 사용자에게는 누구보다 친숙합니다

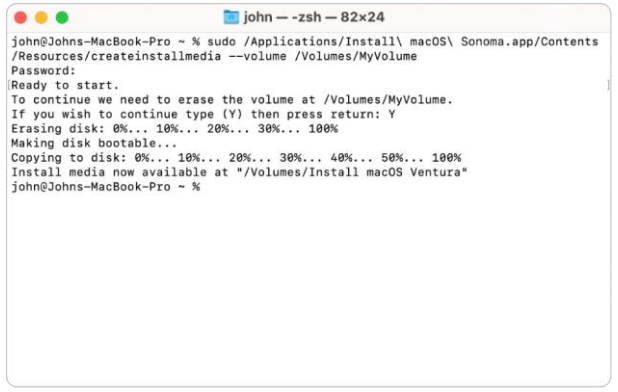
✓아마도 여러분들께서 연구용 혹은 업무용 서버를 제공받는다면 높은 확률로 이 환경으로 제공받을 수 있습니다

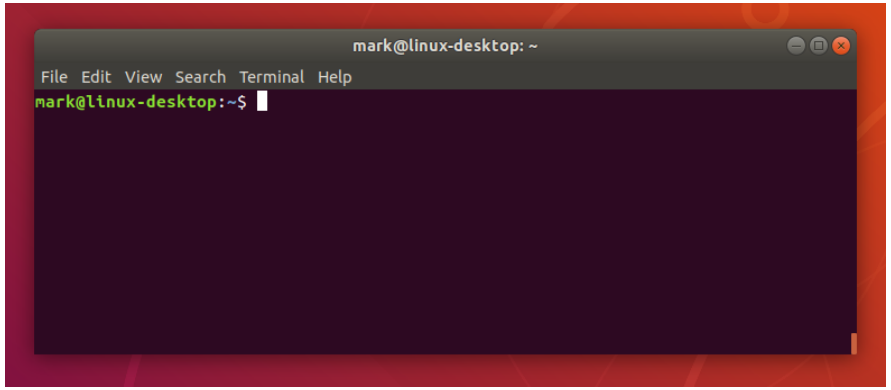
✓그래서 Ubuntu Server를 선택합니다

Ubuntu Server를 설치하고 비교해봅시다

- 현재 사용중인 운영체제와 비교해보는 것입니다
- 각 운영체제는 대화형 커맨드라인 프로그램인 터미널 환경을 제공합니다
- 이를 중심으로 비교해봅니다







Ubuntu Server 설치 이미지를 통한 설치

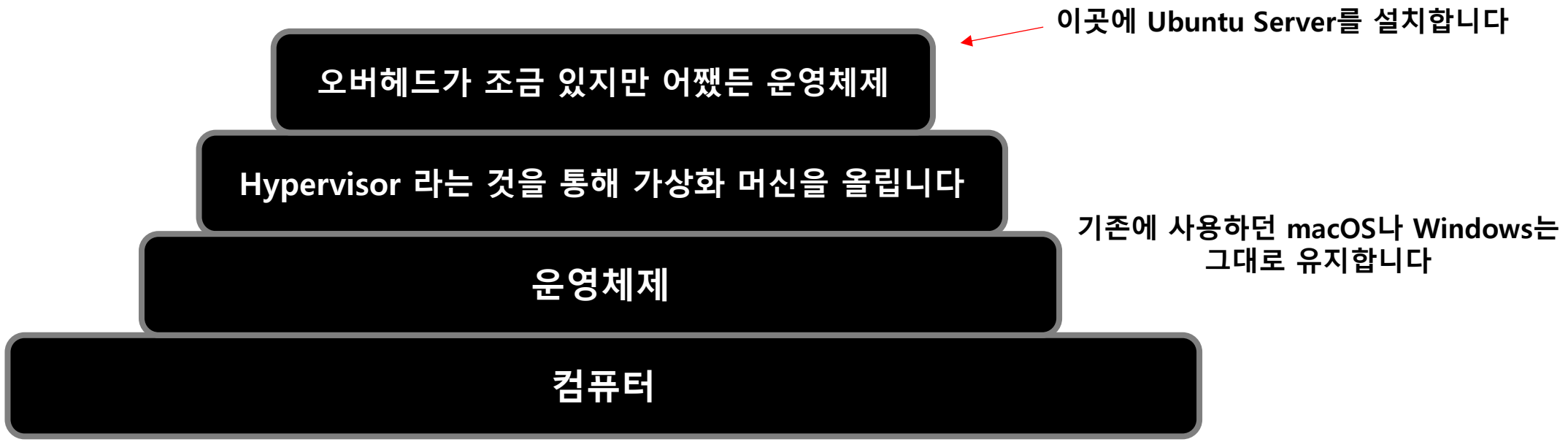
- 출시년월이 버전인 특징아닌 특징
- 매년 4월, 10월에 배포하며 짝수년도에 장기 지원 버전을 배포

- EOL
  - End Of Life: OS의 생명주기, 업데이트 지원 만료 주기
  - 장기 지원 버전 기준 배포후 5년 후에 만료
  - 24.04LTS를 받으면 2029-04에 EOL

- 가장 최신의 장기 지원 버전은 24.04LTS
  - 24.04LTS
    - 2024년 04월 배포한 장기 지원 버전
    - 내년 4월에는 26.04LTS가 나올 것입니다

가상화

- 사용하던 운영체제를 지우고 새로 설치하기는 어렵습니다
- 그래서 가상화를 사용합니다



WSL

- 윈도우에서 돌릴 수 있게 Ubuntu를 변환한 것입니다
- 간편하게 설치할 수 있으나 Original Ubuntu Server 와 약간 다른 점이 있습니다
- 그래서 WSL 보다는 Original Ubuntu Server 이미지를 사용해보겠습니다

Hypervisor

- 가상화 운영체제를 설치할 수 있도록 머신을 가상화 해주는 프로그램

- 아래의 두 이미지는 같은 것을 사용합니다
  - 가상화 운영체제에 사용되는 설치 이미지
  - 가상화 하지 않은 운영체제에 사용되는 설치 이미지

- 따라서 가상화를 통해 운영체제를 설치했을 때 원래의 운영체제 그대로와 비슷하게 사용할 수 있겠습니다

Hypervisor for MacOS

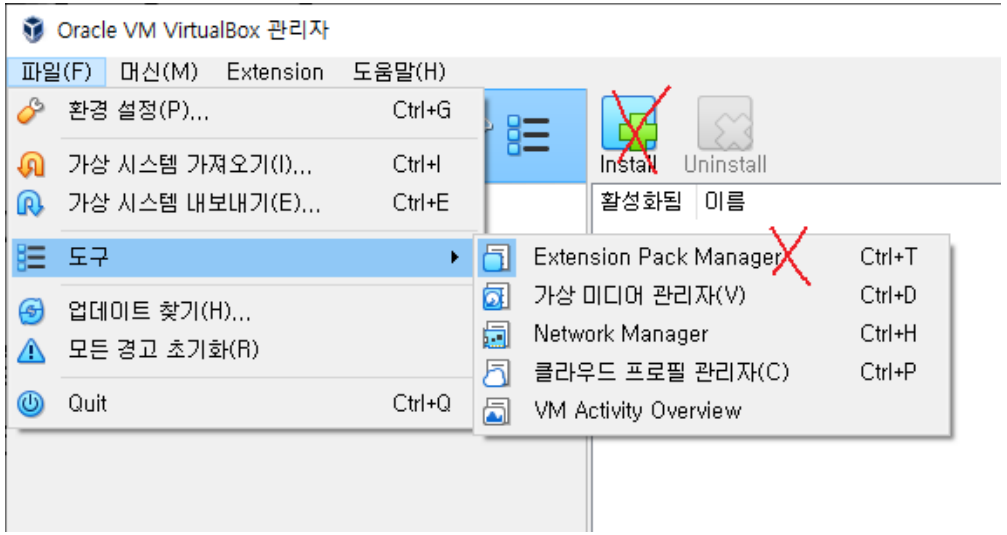
- Hypervisor for MacOS 로 검색해봅니다
- UTM 이라는 것이 나올 것입니다 

Hypervisor for MS Windows

- Virtualbox: 대표적인 무료 하이퍼바이저
- https://www.virtualbox.org/wiki/Downloads 에 들어가봅시다
- Windows hosts 를 선택합니다

Virtualbox 설치시 주의사항

- Virtualbox extension pack는 라이선스 이슈가 발생할 수 있으므로 설치하지 않도록 주의합시다
- https://www.virtualbox.org/wiki/Licensing_FAQ 의 “free license for personal, educational or evaluation use” 을 참조합시다

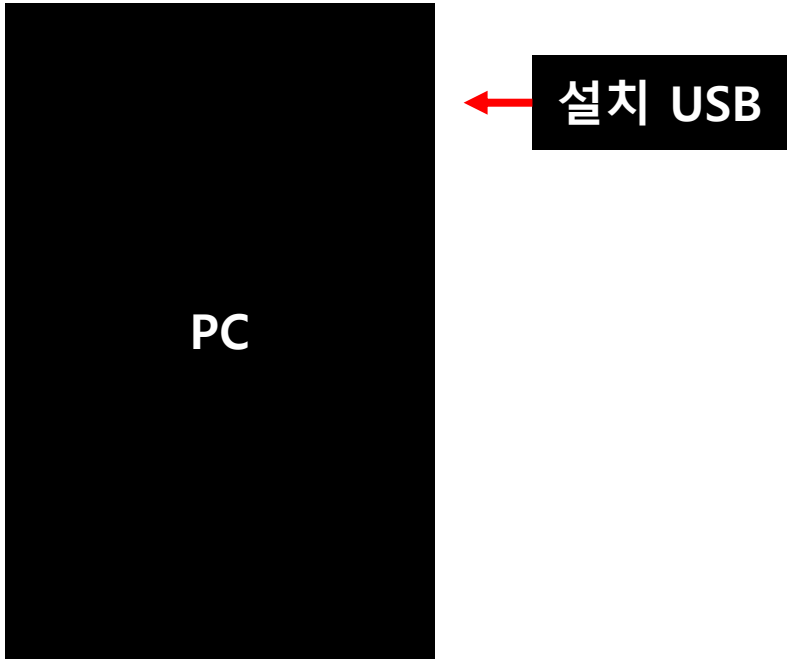


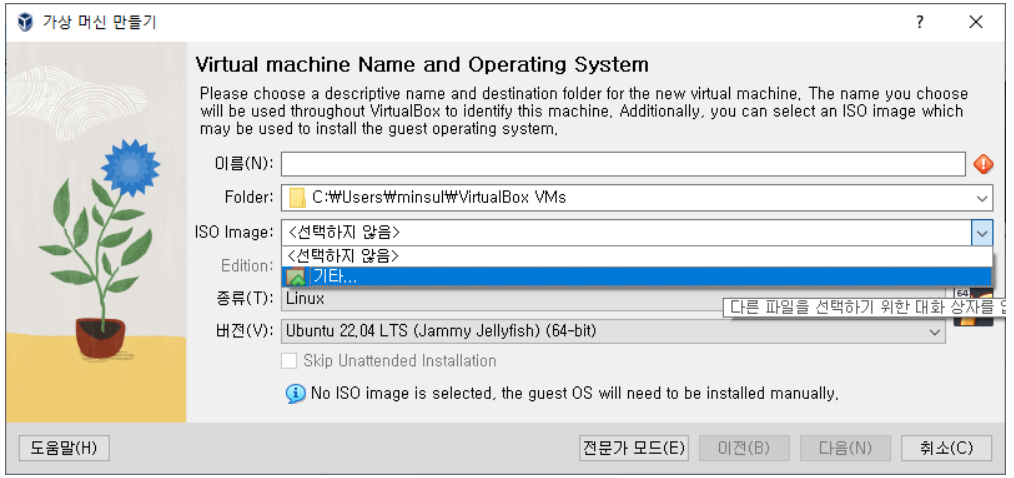
Hypervisor 프로그램을 열면 보이는 것

- 가상머신을 관리하는 프로그램이기 때문에 가상머신 만들기, 가상머신 시작(부팅) 하기는 반드시 존재하겠습니다
- 그리고 가상머신의 하드웨어 설정하는 부분이 있겠습니다
- 아래는 하나의 가상머신이 만들어진 상태 입니다
- 우리는 새로 만들기부터 시작합니다

설치 USB를 PC에 연결하듯이
설치 이미지를 Hypervisor에 연결합니다

- 다운로드 받은 Ubuntu server 22.04 이미지(iso) 파일을 지정합니다
- 그리고 가상머신을 구동하면 그 후로는 실제 운영체제 설치과정과 동일합니다
- 가상머신을 구동하기 전에 다음에 나오는 네트워크 설정을 먼저 진행합니다





Ubuntu server 설치화면

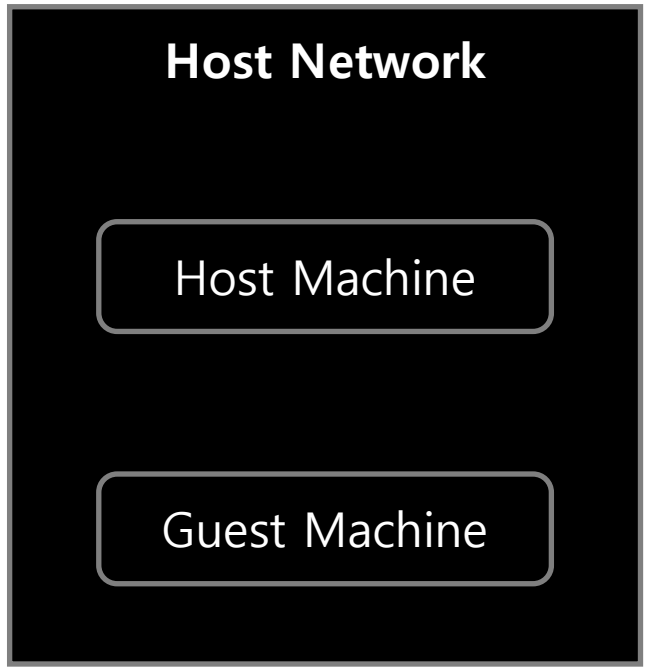
- 많이 선택하는 값으로 기본값이 선택되어 있기 때문에 Next를 누르는 것만으로 운영체제 설치가 완료 됩니다
- 선택되는 값들은 아래와 같습니다
- 그래서 인프라에 대한 기본적인 사항을 확인하고 설치과정으로 넘어가도록 하겠습니다

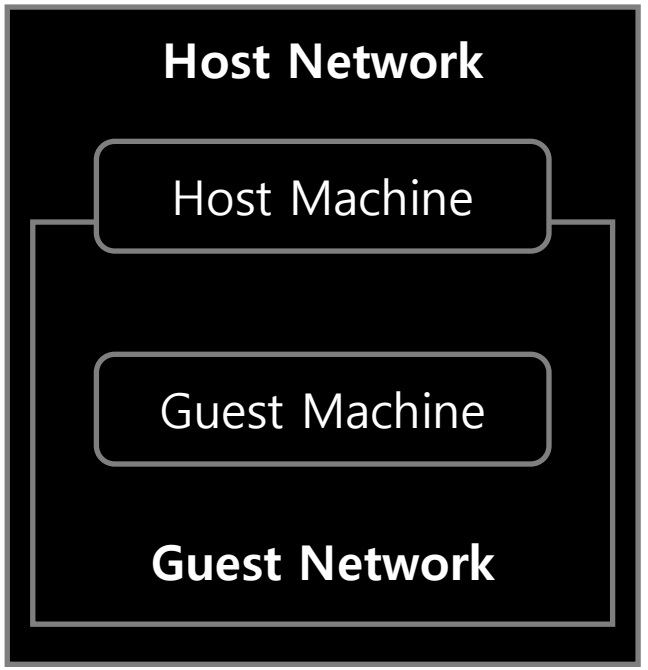
✓네트워크(ip) 설정(자동 설정, 수동 설정)
✓디스크 공간 설정(어느 드라이브를 사용할지)
✓계정 설정(계정 이름, 패스워드: 중요하므로 반드시 기억합니다)
✓부가 설치 패키지 설정

HostOS와 GuestOS간의 네트워크 연결

- HostOS는 Ubuntu Server 설치 전에 원래 설치되어있던 OS를 의미합니다
- Host Network 는 HostOS가 설치된 Host Machine이 소속된 네트워크 입니다
  - 간단히 말해서 노트북 랜선 연결된 네트워크
- GuestOS는 지금 설치하는 Ubuntu Server를 의미합니다

Bridged Mode

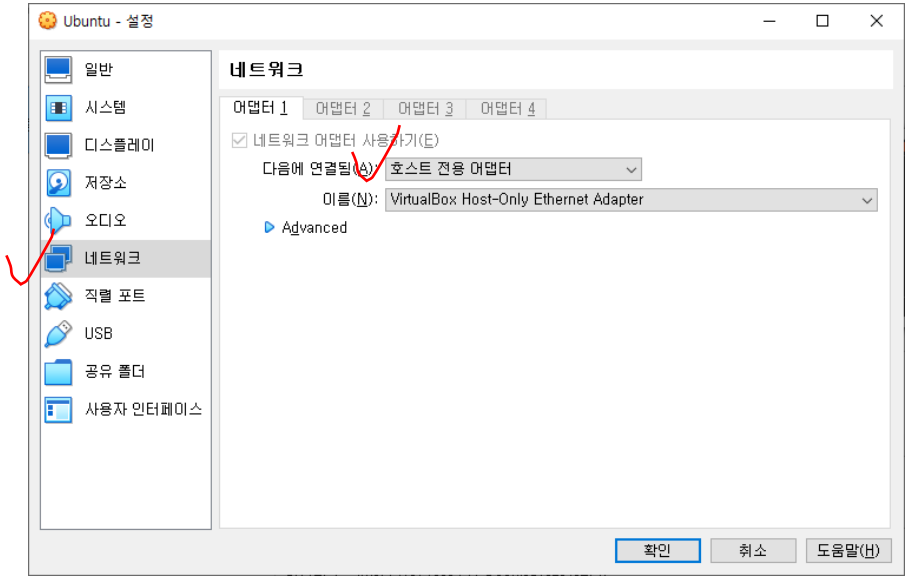




NAT Mode

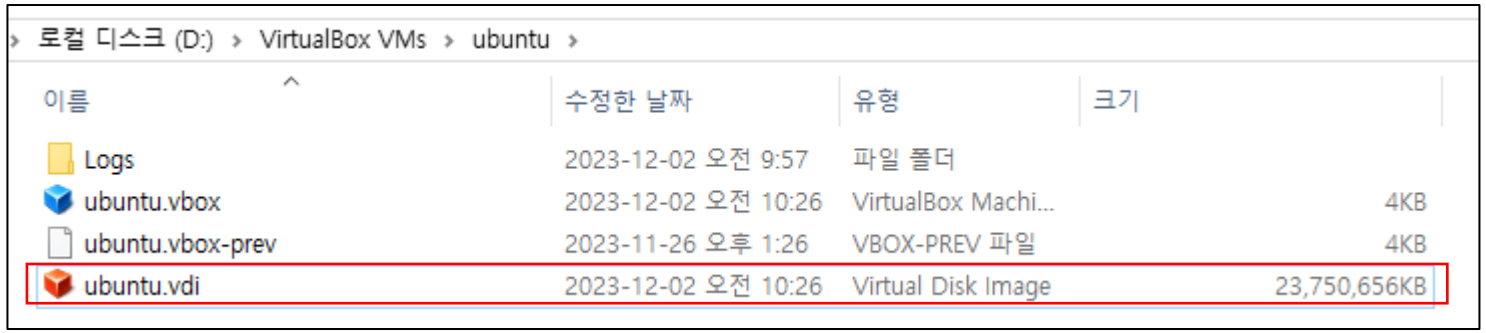
가상머신 네트워크 설정

- UTM에는 이 설정이 필요하지 않습니다
- Virtualbox 사용중이신 경우 설정 > 네트워크 > 호스트 전용 어댑터 설정으로 바꾸어 줍니다
- 이는 Host Machine와 Guest Machine 간의 전용 네트워크로 연결함을 의미합니다



디스크 볼륨과 파일시스템

- Host OS의 공간에는 아래와 같은 큰 용량의 파일이 생깁니다
- 이 파일은 Guest OS(가상머신) 내에서 하나의 SSD 또는 하드디스크, 혹은 USB메모리 같은 것으로 동작한다고 보시면 되겠습니다
- 이러한 SSD 또는 하드디스크, 혹은 USB 메모리 같은 것들을 PC에 연결했을 때 보이는 것은 MS Windows와는 비슷한 부분도 있고 다소 차이가 있는 부분도 있습니다



22GB 정도 되는 가상 SSD

Ubuntu Server의 파일구조와 MS Windows의 파일구조

- 혹시 안드로이드 휴대폰을 가지고 계신다면 갤러리에서 사진을 하나 선택해서 상세정보를 보시면 파일의 경로를 확인하실 수 있습니다
- 아이폰의 운영체제인 iOS는 Unix-Like OS 입니다

- Ubuntu Server, 안드로이드, iOS 모두 Unix-Like OS로서 파일 구조상 비슷한 부분이 있습니다
- 그래서 확인할 수 있는 부분은 아래와 같습니다

MS Windows 의 파일 구조	Linux, Unix-Like OS 의 파일 구조
물리 디스크 하나를 여러 개의 파티션으로 나눌 수 있습니다	물리 디스크 하나를 여러 개의 파티션으로 나눌 수 있습니다
파티션을 파일시스템으로 포맷해서 드라이브로 사용합니다	파티션을 파일시스템으로 포맷해서 사용하는데 드라이브라는 표현은 하지 않습니다
각 드라이브는 고유 문자가 있고 루트 드라이브 개념은 없습니다	그 대신 이 중에 하나를 루트 파일시스템(/) 에 붙입니다
새로운 드라이브는 새로운 문자를 할당하여 사용합니다	새로운 드라이브는 루트 파일시스템 하위 경로에 붙습니다(어디든 가능 합니다)
그래서 USB 드라이브를 붙이면 E: F: 와 같은 새로운 문자가 할당 됩니다	그래서 USB 드라이브를 붙여도 별도의 루트 경로가 생기지는 않고 /USB 와 같이 붙습니다

이제 설치를 진행하고 터미널을 탐방할 때입니다

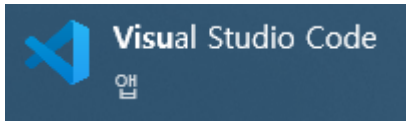
- 운영체제의 터미널을 보는 방법은 아래와 같이 크게 2가지 입니다

✓운영체제가 설치된 PC의 모니터를 보는 방법
- 가상 운영체제의 모니터는 가상머신을 시작했을 때 보시는 그 화면입니다
- GUI 환경을 제공하는 Host OS와의 소통이 원활하지 않습니다

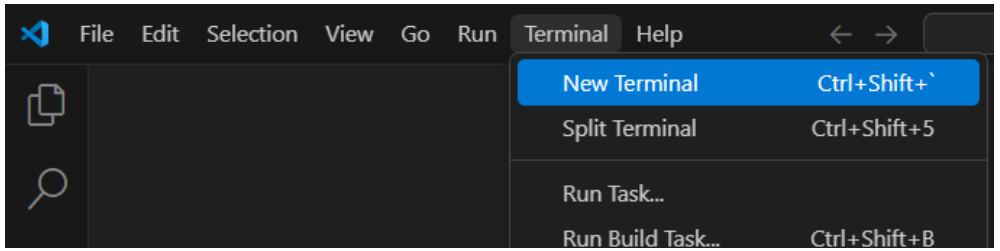
✓운영체제에 원격 접속 하는 방법
- 원격 이라는 것은 꼭 멀리 떨어져 있어야 원격은 아닙니다
- Host OS에서 Guest OS(가상 운영체제)로 연결하는 것도 원격 접속에 해당합니다
- 개발도구에서 원격 개발환경을 제공할 때 편리하므로 이 방법으로 진행합니다
- 개발도구로 넘어가겠습니다

개발도구

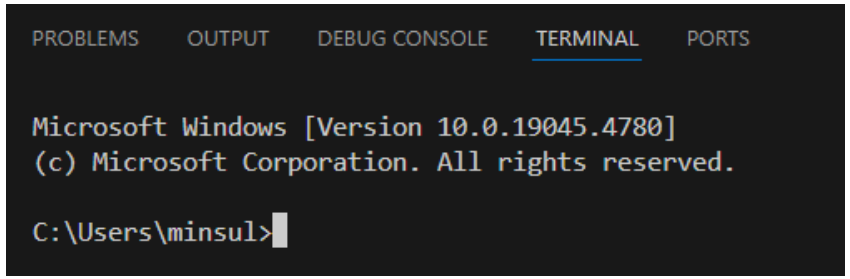
- 여기서 python 프로그래밍도 하고 각종 프로그램 설치도 할 것입니다
- 터미널 원격 접속도 여기서 진행할 것입니다
- 그래서 개발도구를 설치합니다
- 무료로 설치할 수 있는 Visual Studio Code를 설치합시다
- 소스코드 에디터 이지만 터미널 접속 환경 및 원격 개발 환경도 제공합니다



이러한 프로그램을 설치하고 실행합니다



Terminal > New Terminal 을 선택합니다



이 때 보시는 터미널은 HostOS의 터미널 입니다

HostOS에 설치된 Visual Studio Code로부터 GuestOS인 Ubuntu Server에 접속합니다

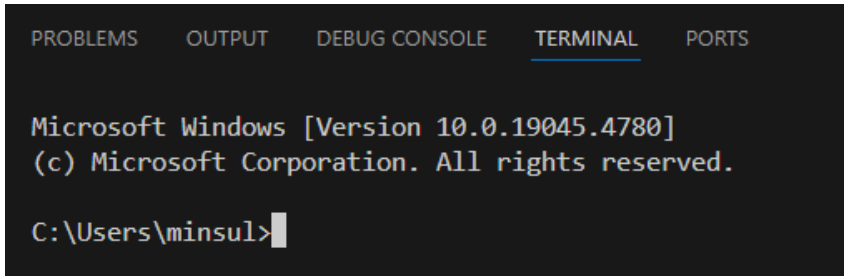
- 원격 접속에는 아래와 같은 요소가 필요합니다
- 발신지 주소와 목적지 주소
- 이것은 우리가 사용하는 대부분의 컴퓨터간 통신에서 동일하니 짚고 넘어가도록 하겠습니다

목적지의 IP	목적지의 Port	발신지의 IP	발신지의 Port
GuestOS의 IP GuestOS도 독립적인 IP를 갖습니다 슬라이드를 되돌려서 HostOS와 GuestOS간의 네트워크 연결을 다시 확인해봅시다	하나의 OS에서 여러가지 서비스를 구동하기 때문에 서비스를 구분하기 위해 포트번호를 지정합니다 발신지측에서 원격 접속 명령시 명시적으로 알고 있어야 합니다 우리가 사용할 번호는 22 입니다	HostOS의 IP 여러분께서 사용하시는 노트북의 IP 입니다 따라서 노트북의 IP 정보를 보면 되겠습니다	Port 번호는 랜덤지정 목적지에서 발신지로 되돌아오는 통신인지 구분하기 위해 발신지의 Port 번호도 필요합니다 그 번호는 목적지에서 알고만 있으면 되고 널리 알려질 필요는 없습니다 그래서 통신 프로그램이 무작위로 하나 골라서 지정합니다

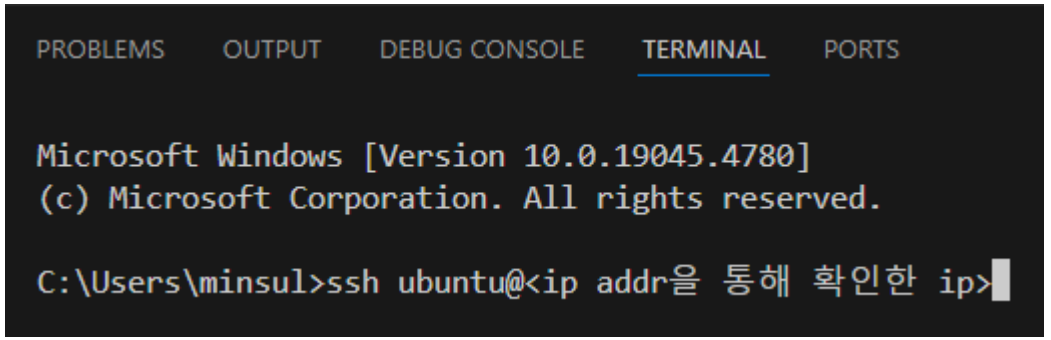
- 마치 편지를 보낼 때 받는 사람과 보내는 사람의 우편번호와 이름을 적는 것과 비슷합니다
- 아직 확인되지 않는 것은 GuestOS의 IP인 목적지의 IP 입니다
- 이를 확인하고 원격 접속 진행하겠습니다

GuestOS의 IP 확인과 원격 접속

- 아직 원격 접속 상태가 아니니까 GuestOS의 모니터에서 확인하겠습니다
- Ubuntu Server를 설치하셨다면 재부팅이 될 것입니다
- 그 후 로그인 메시지가 나올 것입니다
- 지정하셨던 username와 password로 로그인 해주세요
- 그리고 ip addr 을 입력하시면 ip 정보를 확인할 수 있습니다
- 여기까지만 확인하고 다시 Visual Studio Code로 넘어가서 아래의 과정을 진행합니다



여기까지 확인했었습니다



이렇게 명령줄을 입력하고 엔터키를 누르면
발신지 IP는 여러분의 Host Machine의 IP를, 발신지 Port는
무작위로, 목적지 Port는 22로 하는 원격 접속을 진행합니다

달라진 터미널의 모습이 보일 것입니다

- 뭔가 다른 화면이 보일 것입니다
- Host OS가 Ubuntu Server 혹은 리눅스가 아닌 이상 다를 것입니다
- 지금까지 사용하던 운영체제가 아닌 다른 Guest 운영체제에 연결되었음을 의미합니다
- 이 환경에서 앞으로의 과정을 진행하겠습니다