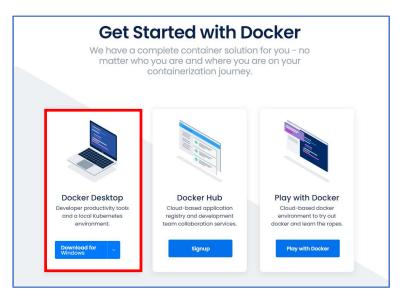
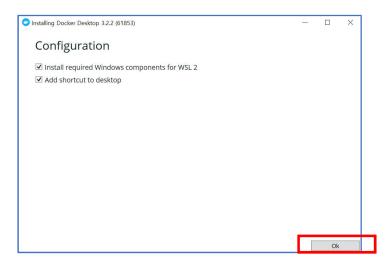
## Installation Docker on Windows 10 WSL2: Ubuntu

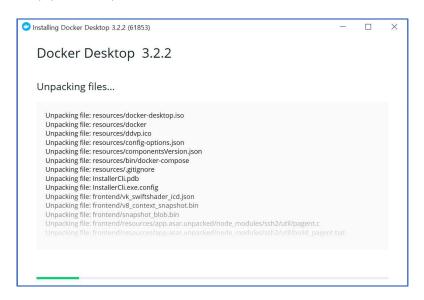
- 1. Prerequisites
  - Windows 10 Pro 이상 빌드 번호는 18362.1049 이상 또는 18363.1049 이상이고, 부 빌드 번호 는 .1049 이상
  - WSL 2
  - Windows Terminal(<a href="https://docs.microsoft.com/ko-kr/windows/terminal/get-started">https://docs.microsoft.com/ko-kr/windows/terminal/get-started</a>)
  - Visual Studio Code for Windows
- 2. Docker Homepage(https://www.docker.com/) 방문한다.
- 3. 화면 우측 상단의 [Get Started] 버튼을 클릭한다. https://www.docker.com/get-started
- 4. 현재 사용자가 Windows를 사용하기 때문에 제일 왼쪽의 [Docker Desktop]의 [Download for Windows] 를 클릭한다.



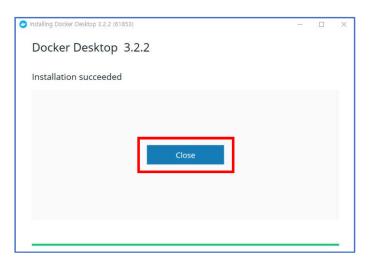
- 5. [Docker Desktop Installer] 파일을 다운로드 후 설치한다.
- 6. 이미 WSL2가 설치되어 있기 때문에 기본적으로 체크되어 있는 내용 확인 후 [OK] 버튼을 누른다.



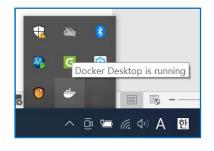
7. 설치가 진행된다.



8. 설치가 성공적으로 마치면 다음과 같은 창이 나온다. [Close] 버튼을 클릭하여 모든 설치 과정을 마무리한다.



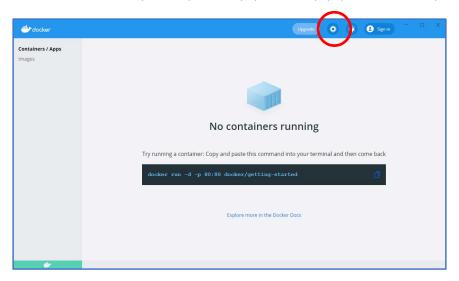
9. 시스템을 Logout 후 다시 Login하면 시스템의 상태표시줄에 **Docker** 아이콘이 보이고 현재 **Docker**가 실행되고 있는 것을 알 수 있다.



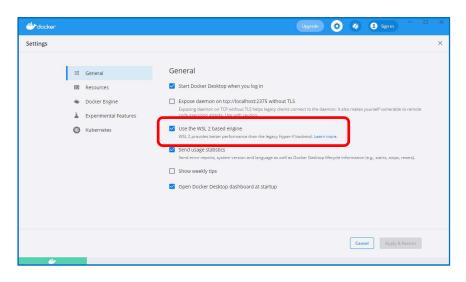


10. 바탕화면의 [Docker Desktop] 아이콘( )을 더블클릭하여 프로그램을 오픈한다.

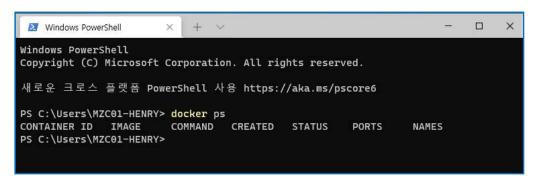
11. 프로그램을 오픈한 뒤 상단의 설정 아이콘을 클릭하여 설정으로 들어간다.



12. WSL2 기반의 엔진을 사용함에 체크되어 있음을 볼 수 있다.



- 13. Windows Terminal을 오픈하여 미리 설치한 WSL 2 Ubuntu를 오픈한다. 실행중인 Docker Image를 검색하기 위해 다음의 명령어를 실행한다.
  - \$ docker ps



14. 현재 WSL2 배포판 설치 목록을 확인한다.

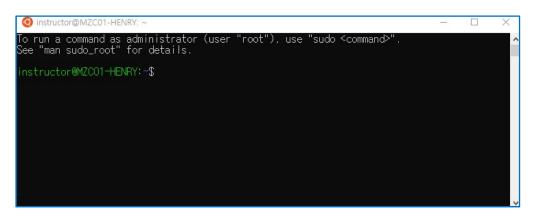
## wsl -l

```
PS C:\Users\MZC01-HENRY> wsl -l
Linux용 Windows 하위 시스템 배포:
Ubuntu-20.04(기본값)
docker-desktop
docker-desktop-data
PS C:\Users\MZC01-HENRY>
```

15. 현재 설치되어 있는 배포판 Ubuntu-20.04로 진입한다.

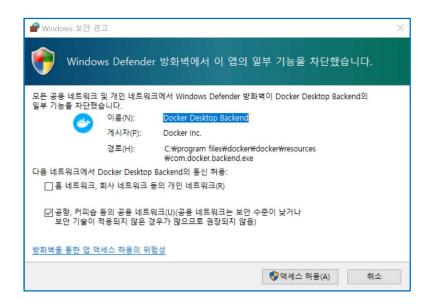
## wsl -d Ubuntu-20.04

또는 [Windows 10 시작 버튼] > [Ubuntu 20.04 LTS]로 Ubuntu 프로그램을 시작한다.

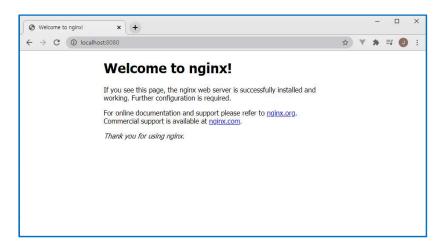


- 16. **Docker에 NginX**를 설치하기 위해 Docker Hub 사이트 https://hub.docker.com/를 방문한다.
- 17. 검색창에서 nginx로 검색한다. Nginx가 이미지가 있는 것을 확인만 한다.
- 18. Ubuntu에서 다음의 명령으로 nginx image를 Docker Hub에서 검색하여 설치하자.
  - \$ docker run -p 8080:80 nginx

```
instructor@MZCO1-HENRY:~$
instructor@MZCO1-H
```



19. 모두 설치가 끝나면 브라우저를 오픈하여 8080포트로 연결해 보자.



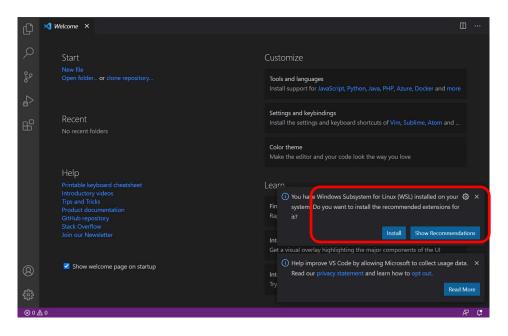
20. 역시 Ubuntu에서도 확인할 수 있다.

172.17.0.1 - - [17/Jun/2021:08:39:05 +0000] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91,0.4472.106 Safari/537.36" "-"
2021/06/17 08:39:05 [error] 31#31: \*2 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost:8080", referrer: "http://localhost:8080/"
172.17.0.1 - - [17/Jun/2021:08:39:05 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://localhost:8080/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.106 Safari/537.38" "-"

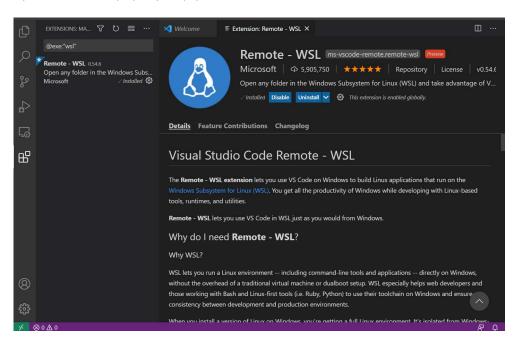
- 21. 이제 Visual Studio Code를 통해 WSL 2 Ubuntu와 연동해 보자.
- 22. Visual Studio Code의 홈페이지(<a href="https://code.visualstudio.com/">https://code.visualstudio.com/</a>)에서 [Search Docs]에 "WSL"로 검색해보자.
  "Use the Windows Subsystem for Linux (WSL)" 페이지가 검색되고 해당 페이지
  (https://code.visualstudio.com/learn/develop-cloud/wsl)로 이동해 보자.



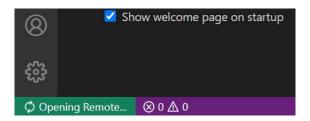
- 23. VS Code에서 WSL을 사용하려면 [Remote WSL extension]을 다운받아야 한다고 되어 있다. VS Code extension에서 [Remote WSL extension] 검색하여 설치하도록 한다.
- 24. 만일 VS Code를 WSL 설치 후에 설치했거나, 그 전에 미리 설치했더라도 WSL 설치 후에 VS Code를 오 픈하면 WSL이 자동인식이 되어 다음과 같이 MarketPlace에서 검색할 필요없이 바로 설치가 가능하다.



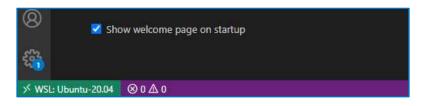
25. 다음 그림은 설치 후 화면이다.



- 26. VS Code의 좌하단의 [Open a Remote Window] 초록색 버튼을 클릭하면 VS Code 상단에 조그만 창이나타나는데, 필자는 여기서 새 창을 열기 위해 [Remote-WSL:New Window]를 선택했다.
- 27. 바로 새 VS Code 창이 열리면서 VS Code 좌하단에 [Opening Remote...]가 나타나서 기존에 설치된 WSL 머신에 접속을 시도하게 된다.



28. 잠시 기다리면 기존에 설치한 WSL: Ubuntu-20.04에 연결이 성공하게 된다.



29. 이제 VS Code에 새 터미널을 오픈해 보자. 아래의 그림처럼 **Windows Terminal**에서 오픈한 커맨드창과 다르지 않다. 즉, 앞으로 Visual Studio Code의 Terminal을 통해서도 얼마든지 **WSL:Ubuntu-20.04**의 개발 작업을 수행할 수 있게 됐다.

