

# Installation Docker on Windows 10 WSL2 : Ubuntu

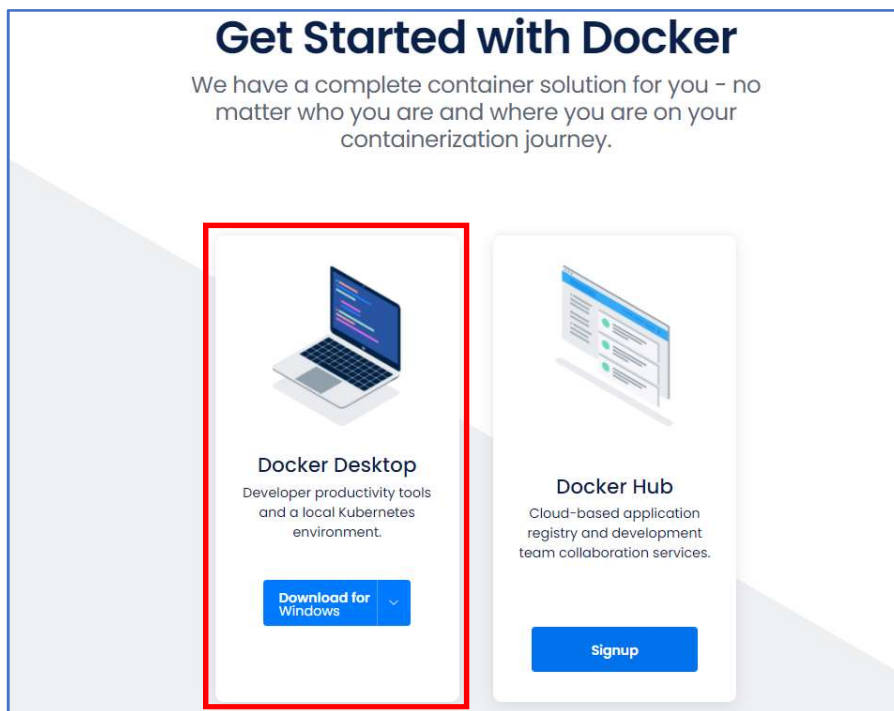
## 1. Prerequisites

- Windows 10 Pro 이상 빌드 번호는 18362.1049 이상 또는 18363.1049 이상이고, 부 빌드 번호는 .1049 이상
- WSL 2
- Windows Terminal(<https://docs.microsoft.com/ko-kr/windows/terminal/get-started>)
- Visual Studio Code for Windows

## 2. Docker Homepage(<https://www.docker.com/>) 방문한다.

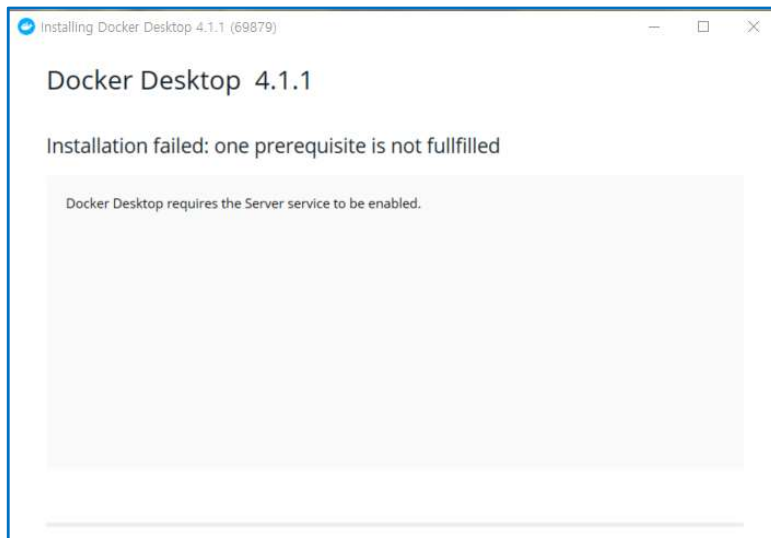
## 3. 화면 우측 상단의 **[Get Started]** 버튼을 클릭한다. <https://www.docker.com/get-started>

## 4. 현재 사용자가 Windows를 사용하기 때문에 제일 왼쪽의 **[Docker Desktop]**의 **[Download for Windows]**를 클릭한다.

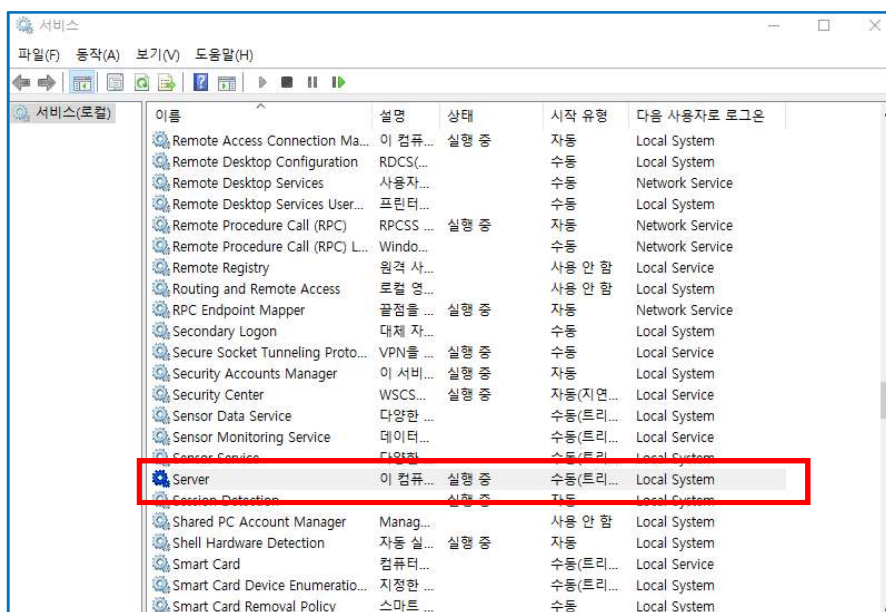


## 5. **[Docker Desktop Installer.exe]** 파일을 다운로드 후 설치한다.

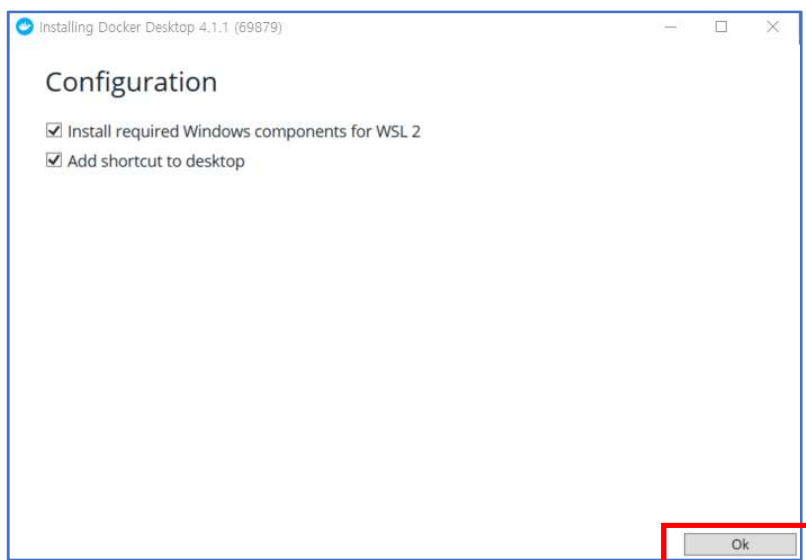
## 6. 만일 Server 서비스가 실행중이 아니면 다음과 같은 오류가 발생한다.



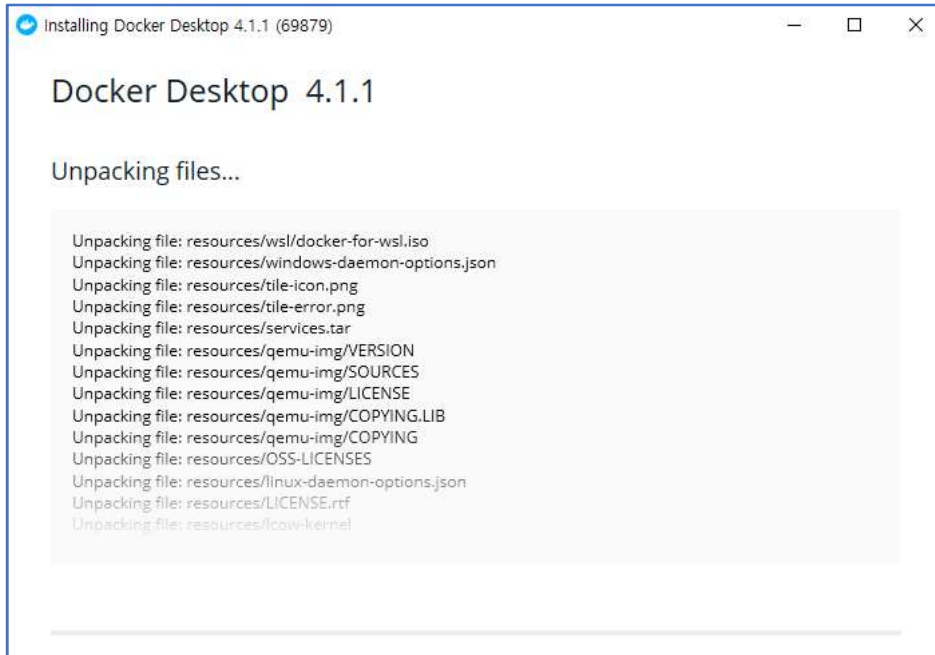
7. Windows 의 [서비스]에 가서 [Server] 서비스를 시작한다.



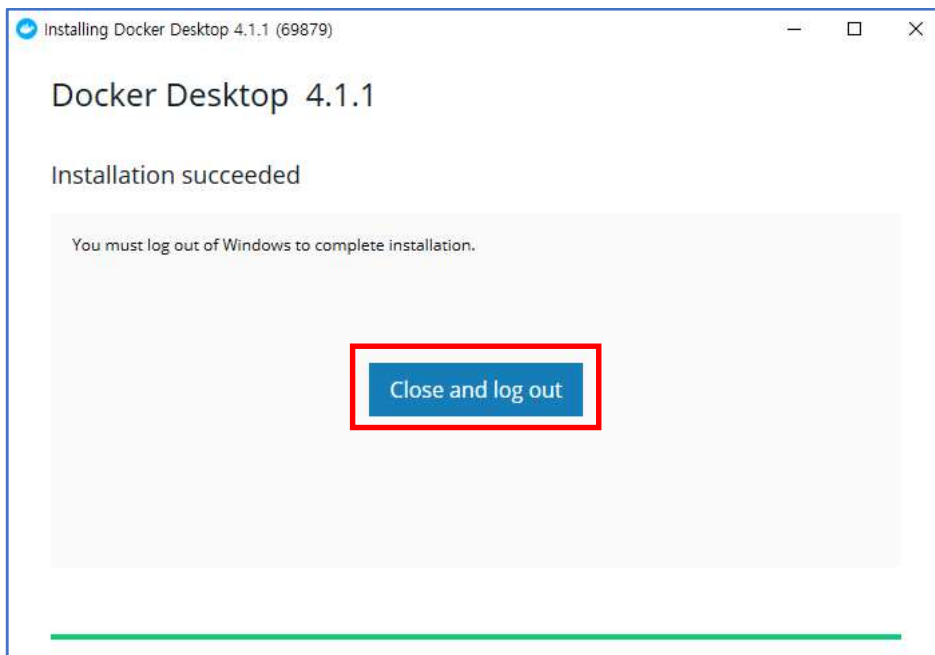
8. 이미 WSL2가 설치되어 있기 때문에 기본적으로 체크되어 있는 내용 확인 후 [OK] 버튼을 누른다.



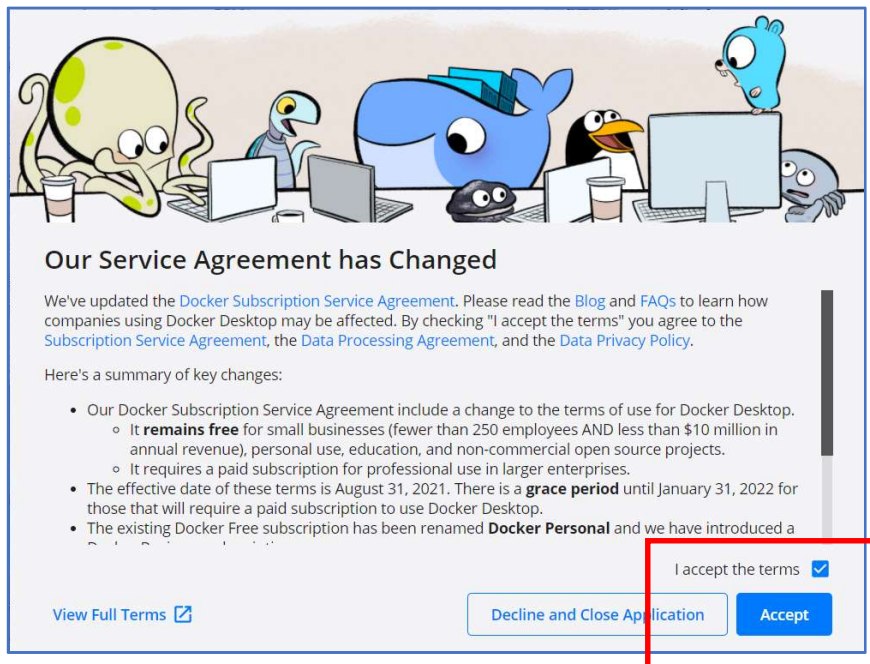
9. 설치가 진행된다.



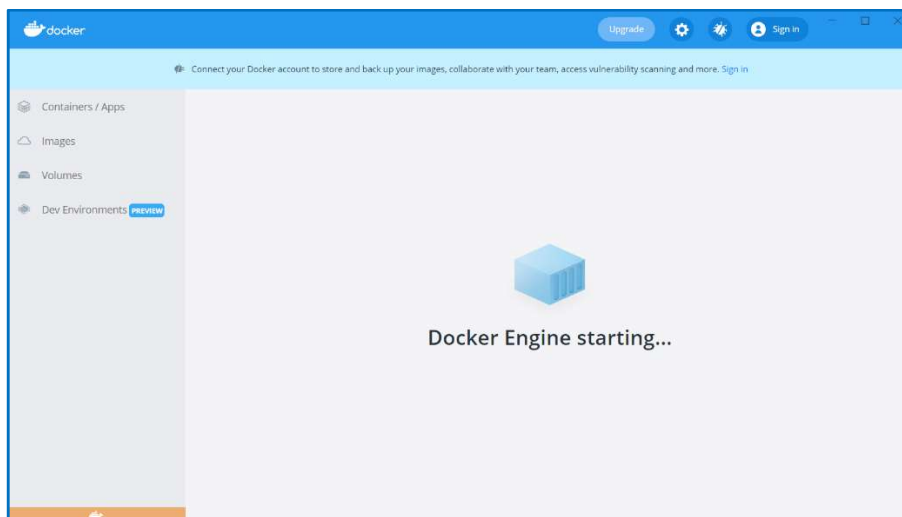
10. 설치가 성공적으로 마치면 다음과 같은 창이 나온다. **[Close and log out]** 버튼을 클릭하여 모든 설치 과정을 마무리한다.



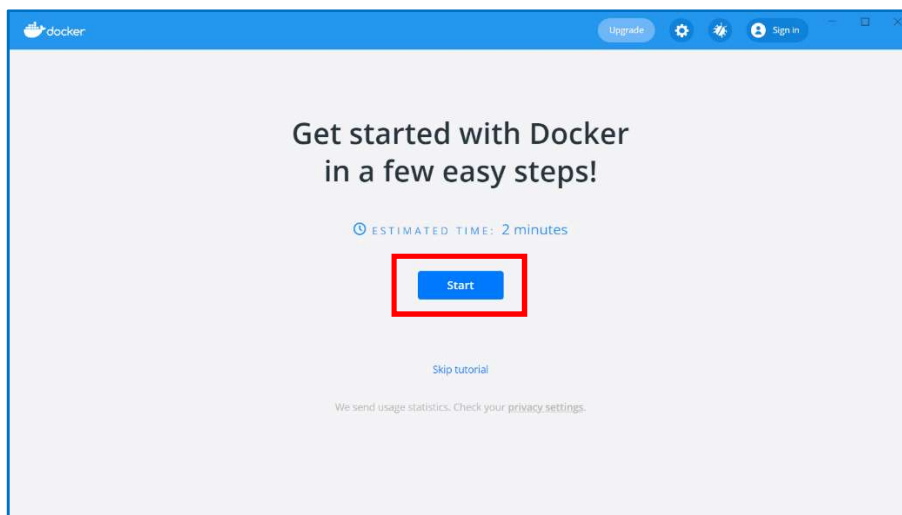
11. 시스템을 Logout 후 다시 Login하면 다음과 같은 화면을 만나게 된다. **[I accept the terms]** 체크 박스에 체크하고 **[Accept]** 파란색 버튼을 클릭한다.



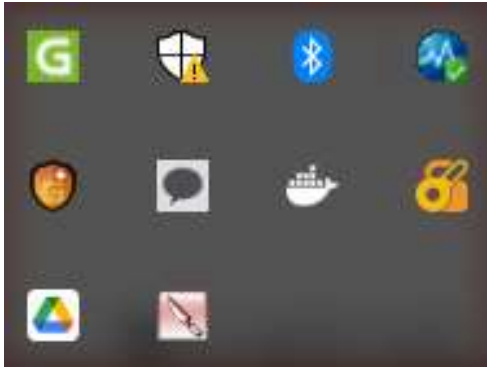
12. [Docker Engine starting...] 화면이 나타난다.



13. Docker 서비스를 시작하기 위해 [Start] 파란색 버튼을 클릭한다.

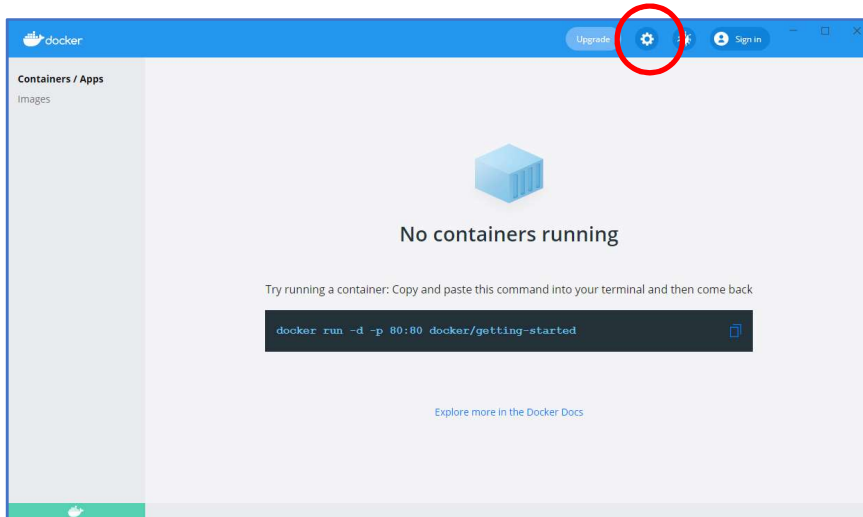


14. 한편, 시스템의 상태표시줄에 보면 **Docker** 아이콘이 보인다.

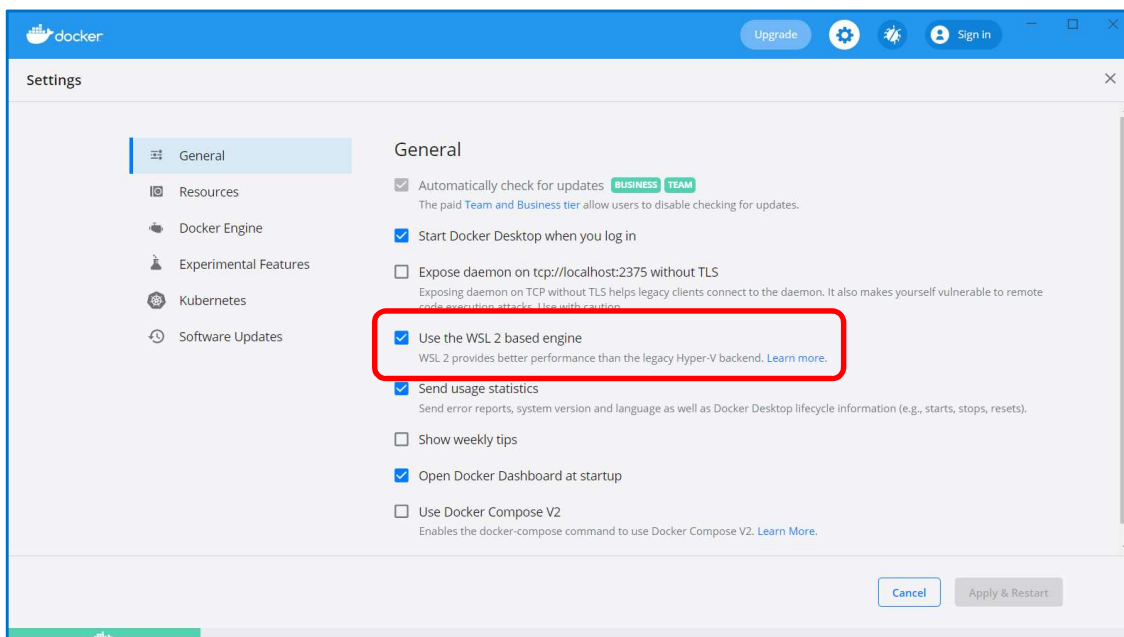


15. 바탕화면의 **[Docker Desktop]** 아이콘(  )을 더블클릭하여 프로그램을 오픈한다.

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



17. **WSL2** 기반의 엔진을 사용함에 체크되어 있음을 볼 수 있다.



18. 다음의 명령을 Windows Terminal 즉 Command 창에서 실행한다.

```
$ docker run -d -p 80:80 docker/getting-started
```

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\WMZC01-HENRY>docker run -d -p 80:80 docker/getting-started
Unable to find image 'docker/getting-started:latest' locally
latest: Pulling from docker/getting-started
540db60ca938: Pull complete
0ae30075c5da: Pull complete
9da81141e74e: Pull complete
b2e41dd2ded0: Pull complete
7f40e809fb2d: Pull complete
758848c48411: Pull complete
23ded5c3e3fe: Pull complete
38a847d4d941: Pull complete
Digest: sha256:10555bb0c50e13fc4dd965ddb5f00e948ffa53c13ff15dcdc85b7ab65e1f240b
Status: Downloaded newer image for docker/getting-started:latest
ee994ba81a141a2d16ff61e0c9bcca74a6516bd874f7ab9065d09501fd1c57b3
```

19. 실행중인 Docker Image를 검색하기 위해 다음의 명령어를 실행한다.

```
$ docker ps
```

```
C:\Users\WMZC01-HENRY>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
ee994ba81a14   docker/getting-started  "/docker-entrypoint.…"   49 seconds ago Up 47 seconds 0.0.0
.0:80->80/tcp   sweet_allen
```

20. 현재 **WSL2** 배포판 설치 목록을 확인한다.

```
$ wsl -l
```

```
C:\Users\WMZC01-HENRY>wsl -l
Linux용 Windows 하위 시스템 배포:
Ubuntu-20.04(기본값)
docker-desktop-data
docker-desktop
```

21. 현재 설치되어 있는 배포판 **[Ubuntu-20.04]**로 진입한다.

```
$ wsl -d Ubuntu-20.04
```

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

```
Instructor@MZC01-HENRY: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

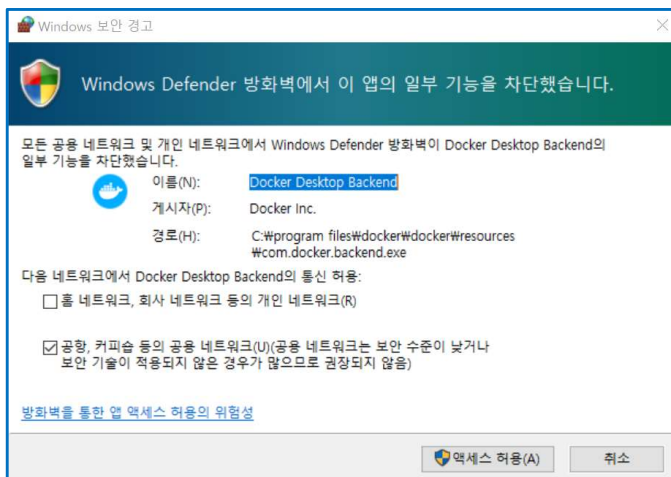
Instructor@MZC01-HENRY:~$
```



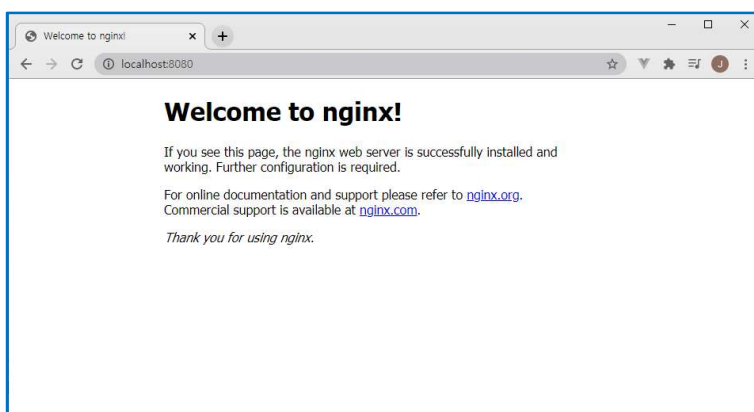
22. Docker에 NginX를 설치하기 위해 Docker Hub 사이트 <https://hub.docker.com/>를 방문한다.
23. 검색창에서 **nginx**로 검색한다. Nginx가 이미지가 있는 것을 확인만 한다.
24. Ubuntu에서 다음의 명령으로 **nginx image**를 Docker Hub에서 검색하여 설치하자.

```
$ docker run -p 8080:80 nginx
```

```
instructor@M2C01-HENRY: ~
instructor@M2C01-HENRY:~$
instructor@M2C01-HENRY:~$ docker run -p 8080:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
Digest: sha256:6d75c99af15565a301e48297fa2d121e15d80ad526f8369c526324f0f7ccb750
Status: Downloaded newer image for nginx:latest
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2021/06/17 08:37:28 [notice] 1#1: using the "epoll" event method
2021/06/17 08:37:28 [notice] 1#1: nginx/1.21.0
2021/06/17 08:37:28 [notice] 1#1: built by gcc 8.3.0 (Debian 8.3.0-6)
2021/06/17 08:37:28 [notice] 1#1: OS: Linux 5.4.72-microsoft-standard-WSL2
2021/06/17 08:37:28 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2021/06/17 08:37:28 [notice] 1#1: start worker processes
2021/06/17 08:37:28 [notice] 1#1: start worker process 31
2021/06/17 08:37:28 [notice] 1#1: start worker process 32
2021/06/17 08:37:28 [notice] 1#1: start worker process 33
2021/06/17 08:37:28 [notice] 1#1: start worker process 34
2021/06/17 08:37:28 [notice] 1#1: start worker process 35
2021/06/17 08:37:28 [notice] 1#1: start worker process 36
2021/06/17 08:37:28 [notice] 1#1: start worker process 37
2021/06/17 08:37:28 [notice] 1#1: start worker process 38
```



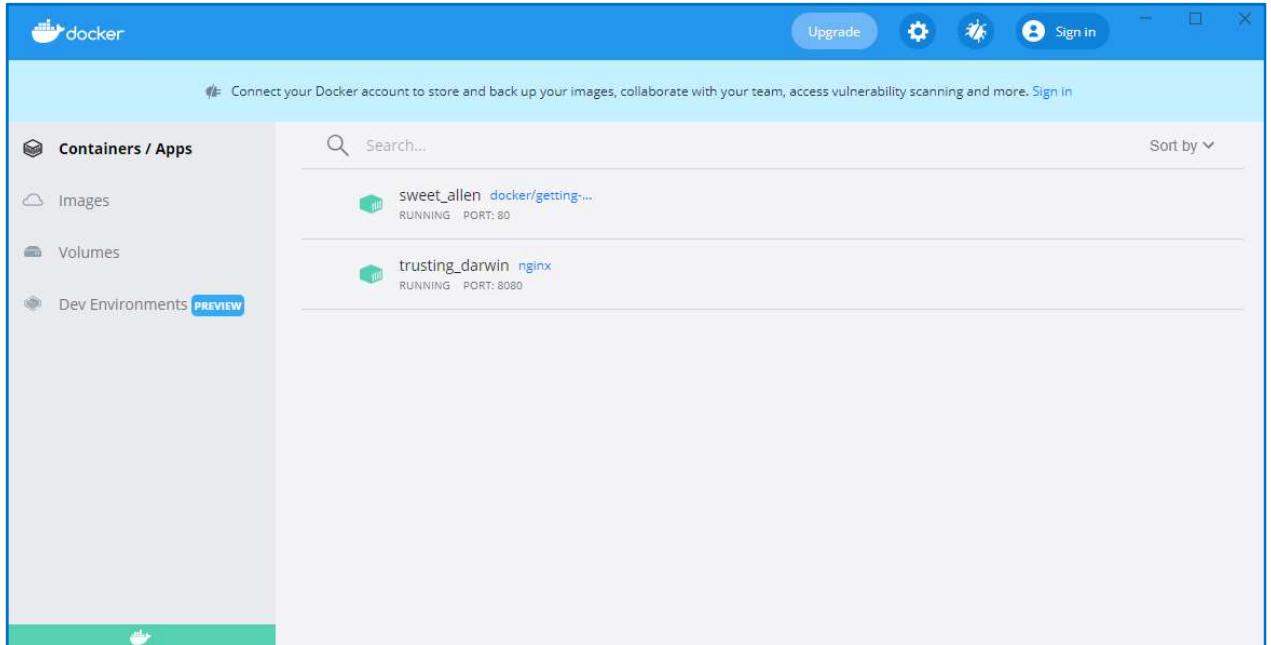
25. 모두 설치가 끝나면 브라우저를 오픈하여 8080포트로 연결해 보자.



26. 역시 Ubuntu에서도 확인할 수 있다.

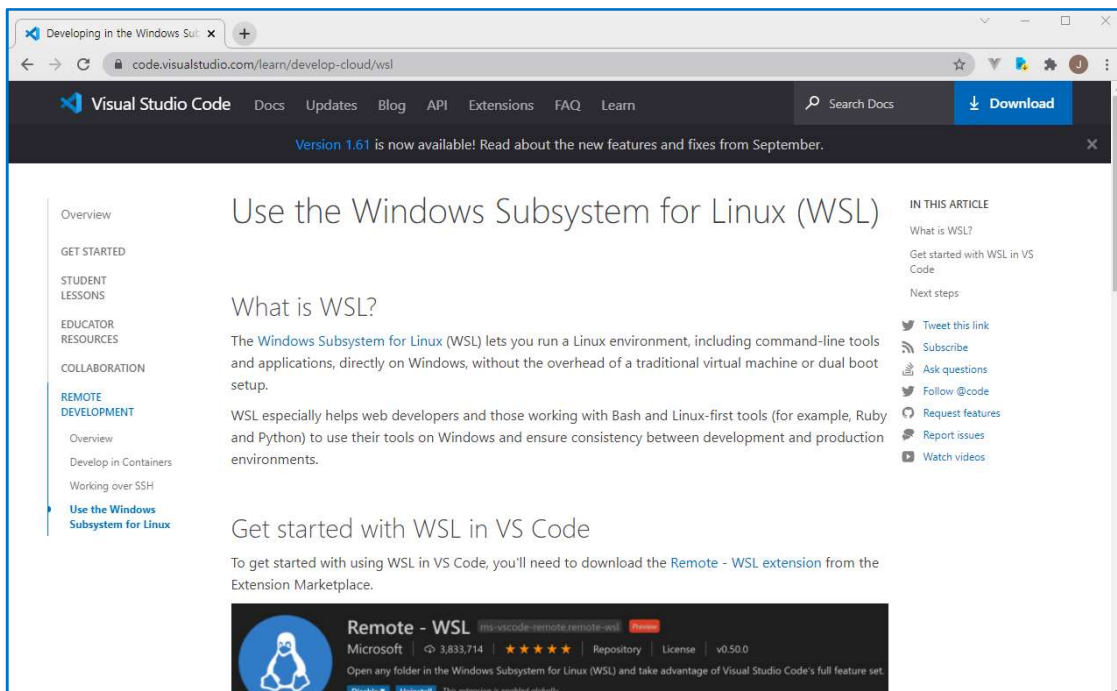
```
172.17.0.1 - - [17/Jun/2021:08:39:05 +0000] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.106 Safari/537.36" "-"
2021/06/17 08:39:05 [error] 31#31: *2 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file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

27. 한편, Docker 창에서도 확인할 수 있다.



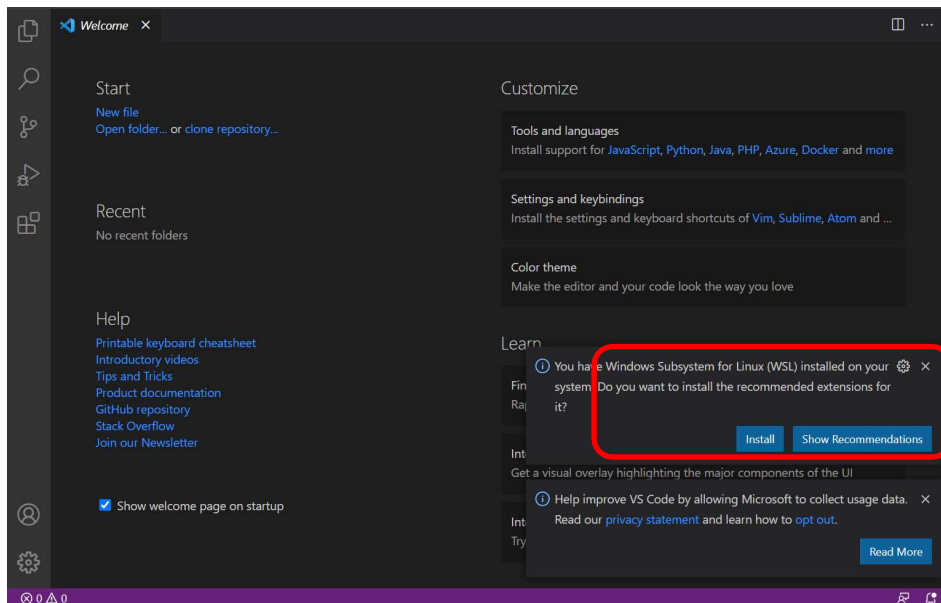
28. 이제 **Visual Studio Code**를 통해 **WSL 2 Ubuntu**와 연동해 보자.

29. Visual Studio Code의 홈페이지(<https://code.visualstudio.com/>)에서 [Search Docs]에 “WSL”로 검색한다. “Use the Windows Subsystem for Linux (WSL)” 페이지가 검색되고 해당 페이지(<https://code.visualstudio.com/learn/develop-cloud/wsl>)로 이동한다.

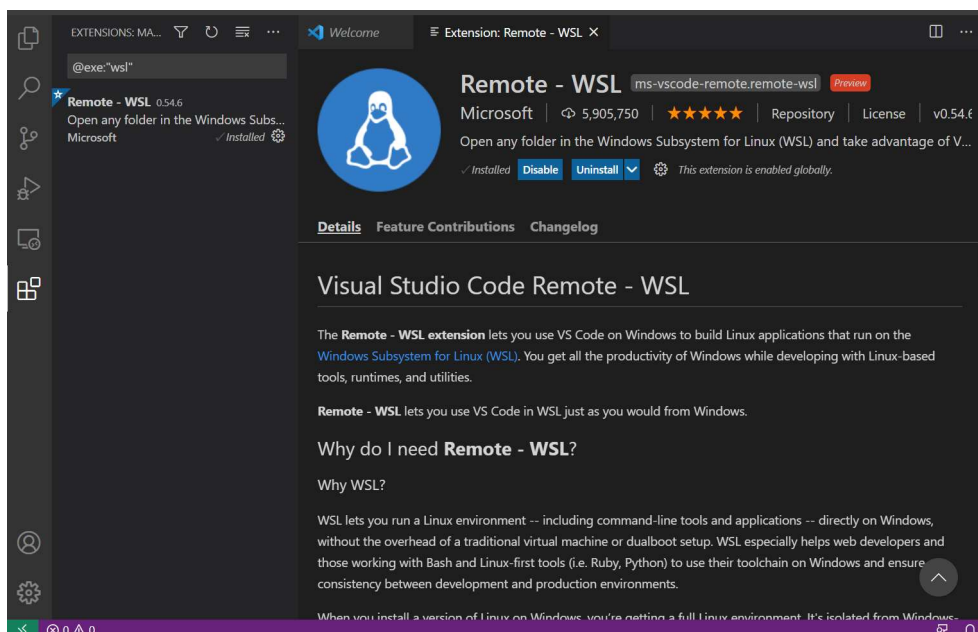




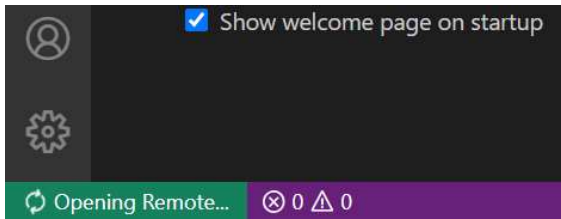
30. VS Code에서 WSL을 사용하려면 **[Remote – WSL extension]**을 다운받아야 한다고 되어 있다. **VS Code extension**에서 **[Remote – WSL]** 검색하여 설치하도록 한다.
31. 만일 VS Code를 WSL 설치 후에 설치했거나, 그 전에 미리 설치했더라도 WSL 설치 후에 VS Code를 오픈하면 WSL이 자동인식이 되어 다음과 같이 MarketPlace에서 검색할 필요없이 바로 설치가 가능하다.



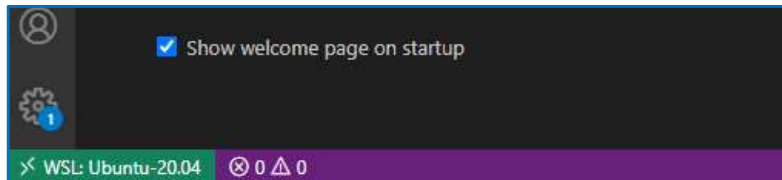
32. 다음 그림은 설치 후 화면이다.



33. VS Code의 좌하단의 **[Open a Remote Window]** 초록색 버튼을 클릭하면 VS Code 상단에 조그만 창이 나타나는데, 필자는 여기서 새 창을 열기 위해 **[Remote-WSL:New Window]**을 입력한다.
34. 바로 새 VS Code 창이 열리면서 VS Code 좌하단에 **[Opening Remote...]**가 나타나서 기존에 설치된 WSL 머신에 접속을 시도하게 된다.



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



36. 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.

