



Docker 설치 및 실행확인

- 1. WSL2를 설치하고 활성화하는 방법(도커 실행환경)
- 2. Docker 설치 (도커설치)
- 3. Docker 설치 확인 및 간단한 nginx 서버 예제 실행해보기(docker행)

단어 설명

WSL2 : 윈도우에서 리눅스 파일 실행하게 해주는 아키텍처

순서

도커 실행환경 → 도커설치 → 도커 동작

1.WSL2를 설치하고 활성화하는 방법(도커 실행 환경)

1. Windows Terminal을 '관리자 권한으로 실행'합니다.
2. PowerShell 탭에서 다음 두 명령어를 실행합니다.

```
$ dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart  
$ dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

3. 두 명령어 모두 '작업을 완료했습니다' 출력으로 종료되었는지 확인합니다.
4. 윈도우를 재부팅합니다.
5. x64 머신용 최신 WSL2 Linux 커널 업데이트 패키지를 다운로드 받아 안내에 따라 설치합니다.
6. Windows Terminal을 열고 다음 명령어를 실행합니다.

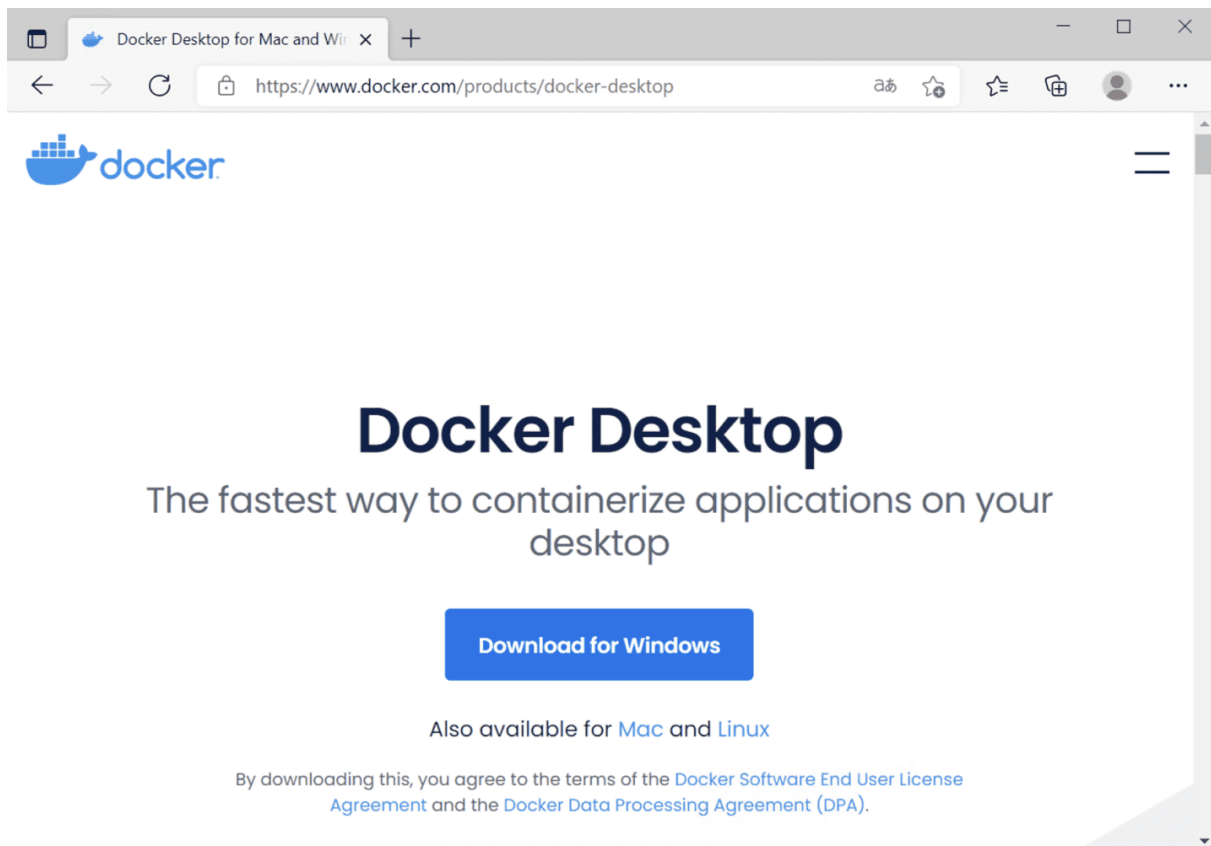
```
$ wsl --set-default-version 2
```

WSL2로 리눅스를 사용하고자 하는 경우, 리눅스 배포판 설치하는 등 추가 설정이 필요하지만, Docker만 사용하는 경우 여기까지만 셋업하면 됩니다. 그럼 이제 Docker Desktop을 설치해보겠습니다.

2. Docker 설치 (도커설치)

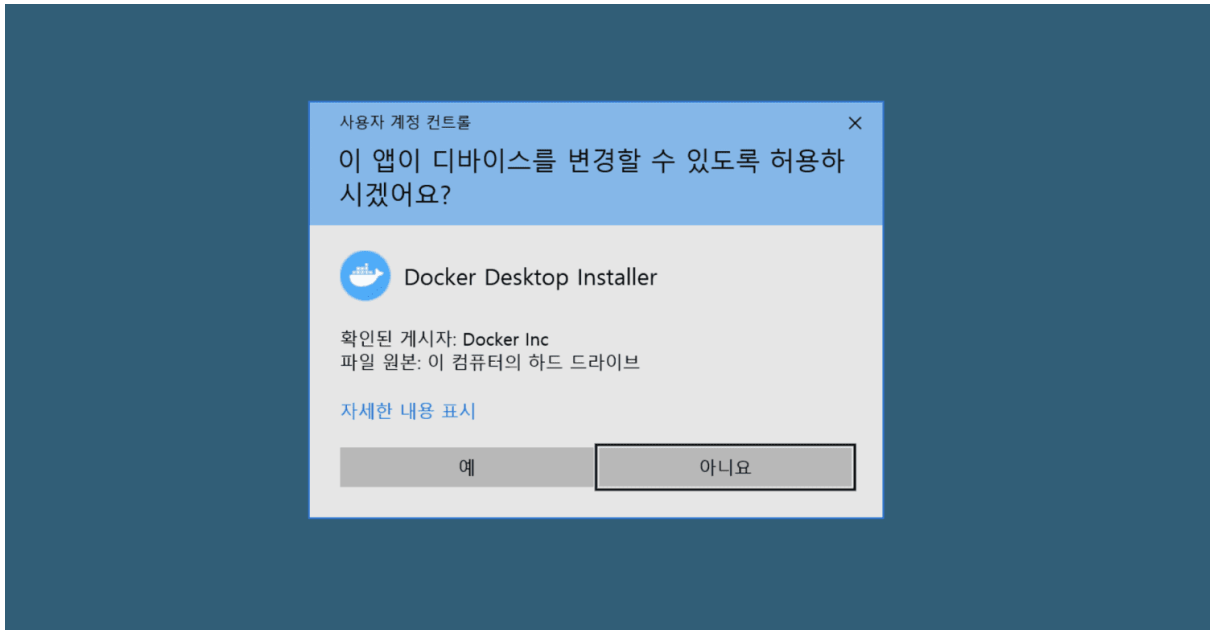
1. 다음 페이지로 이동해서 Download for Windows를 클릭해 Docker Desktop for Windows를 다운로드 받습니다. 여기서는 2021년 6월 현재 최신 버전인 3.5.1을 설치합니다(버전에 따라 조금 다를 수 있으나 설치 과정은 대동소이합니다).

- [Docker Desktop for Mac and Windows](#) | [Docker](#)



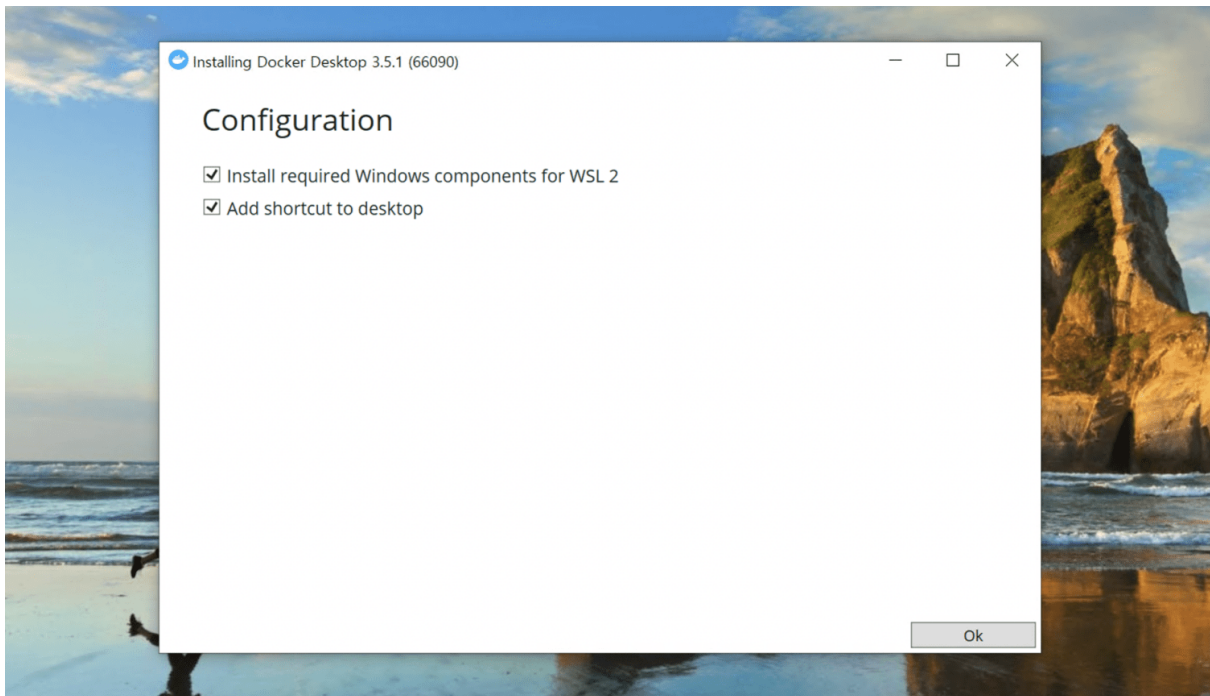
Docker Desktop을 다운로드 합니다

2. 다운로드 받은 `Docker Desktop Installer.exe`를 실행하면 사용자 계정 컨트롤이 나타납니다. 설치를 진행하려면 '예'를 클릭합니다.



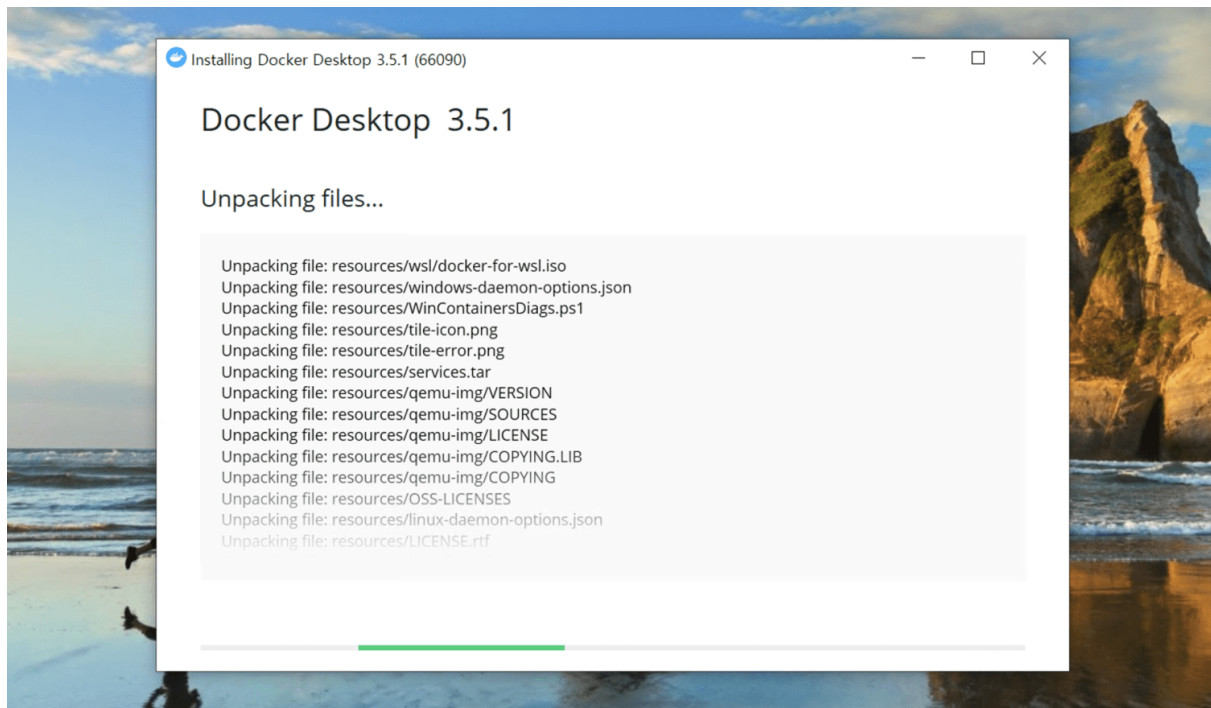
설치를 계속하려면 사용자 계정 컨트롤 팝업에서 예를 클릭합니다

3. 안내에 따라 설치를 진행합니다. 설치 중간에 Configuration이 나타납니다. 둘 다 체크하고 설치를 진행합니다(첫 번째 옵션은 WSL 관련, 2번째 옵션은 바탕화면에 아이콘 추가할지 여부입니다).



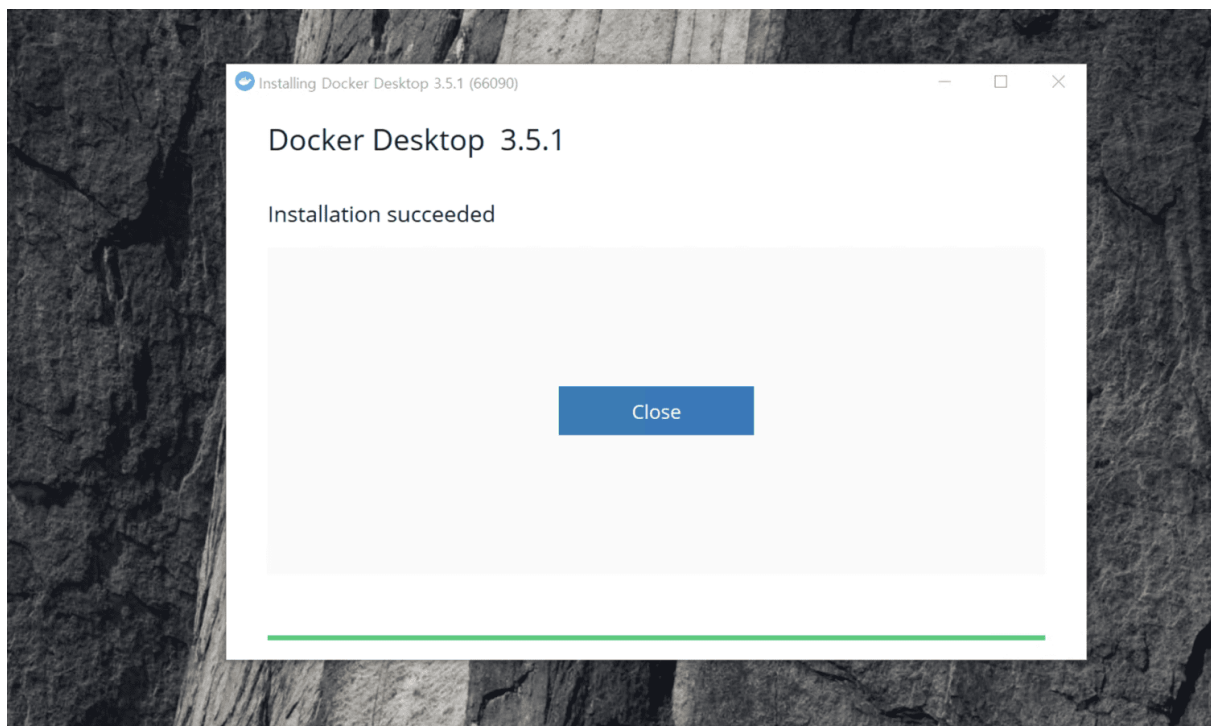
설치 과정에서 필요한 Configuration을 진행합니다

4. Docker Desktop 설치가 진행됩니다. 몇 분 정도 시간이 걸리니 완료될 때까지 기다립니다.



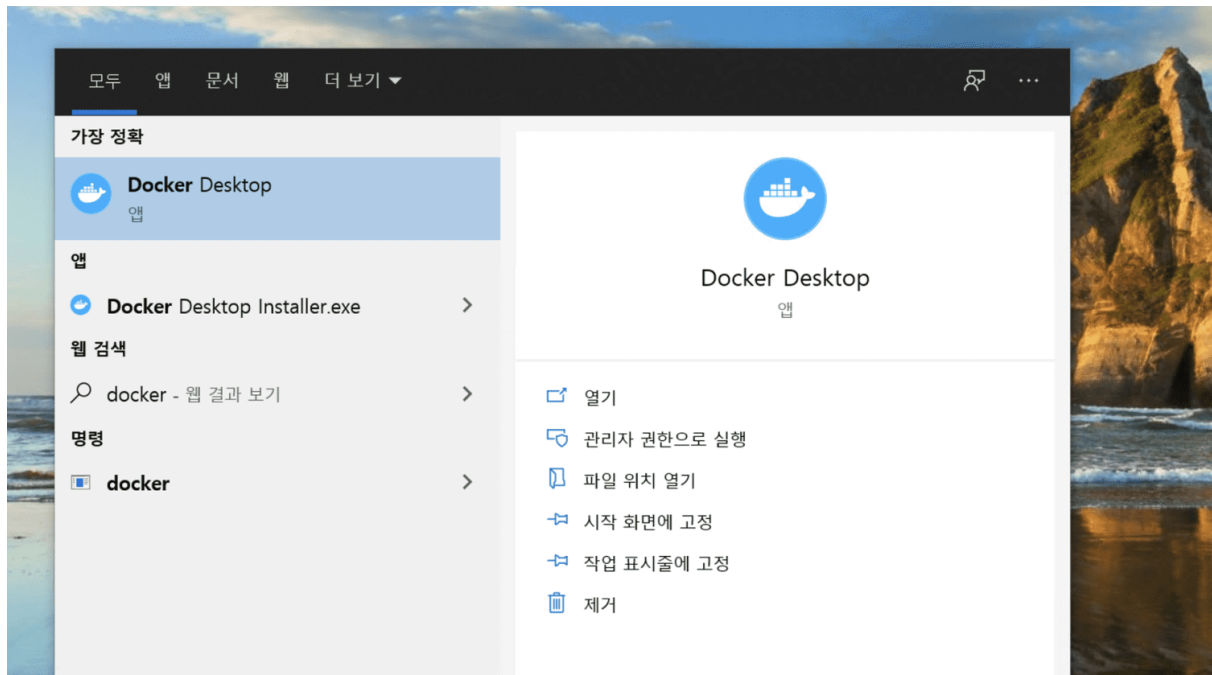
Docker가 설치됩니다. 몇 분 정도 시간이 걸립니다

5. 설치가 끝나면 Installation succeeded 메시지가 나타납니다. 시스템 상태에 따라서 재시작이나 로그아웃을 해야하는 경우도 있습니다. Close 버튼을 클릭해 인스톨러를 종료합니다.



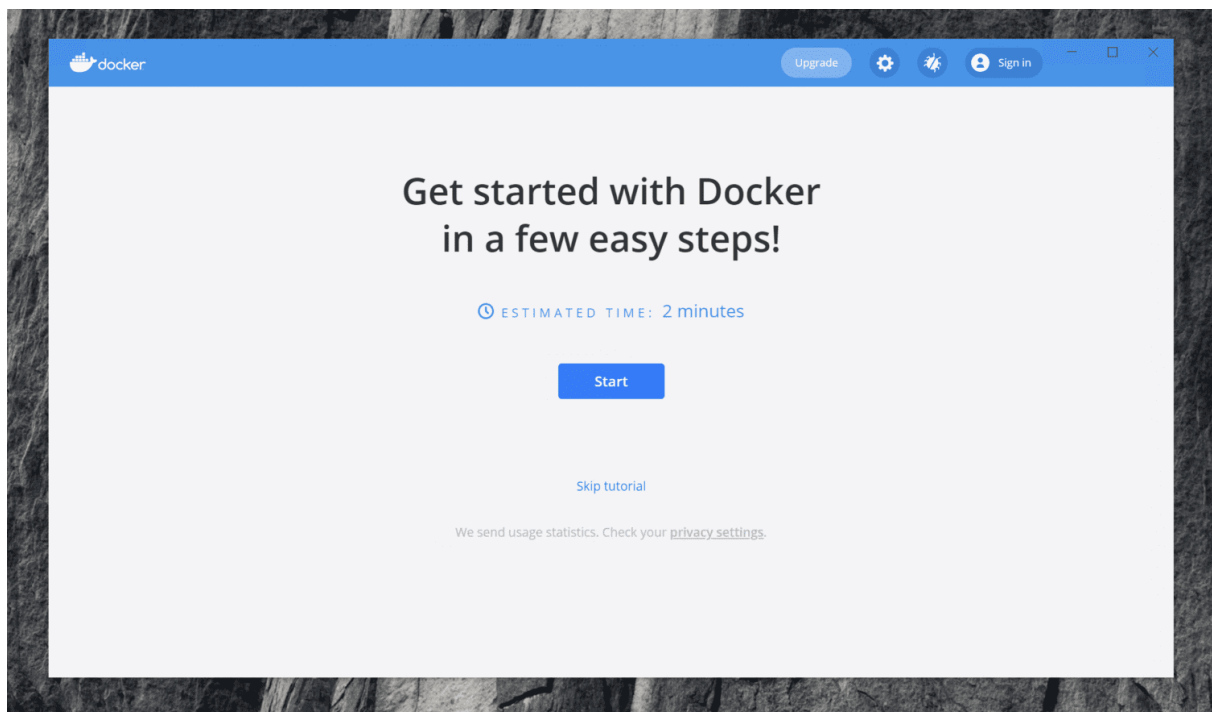
Docker Desktop 설치가 완료되었습니다

6. 이제 바탕화면의 Docker Desktop 아이콘이나 Windows + S로 Docker를 검색해서 Docker Desktop을 실행할 수 있습니다. Docker Desktop을 실행합니다.



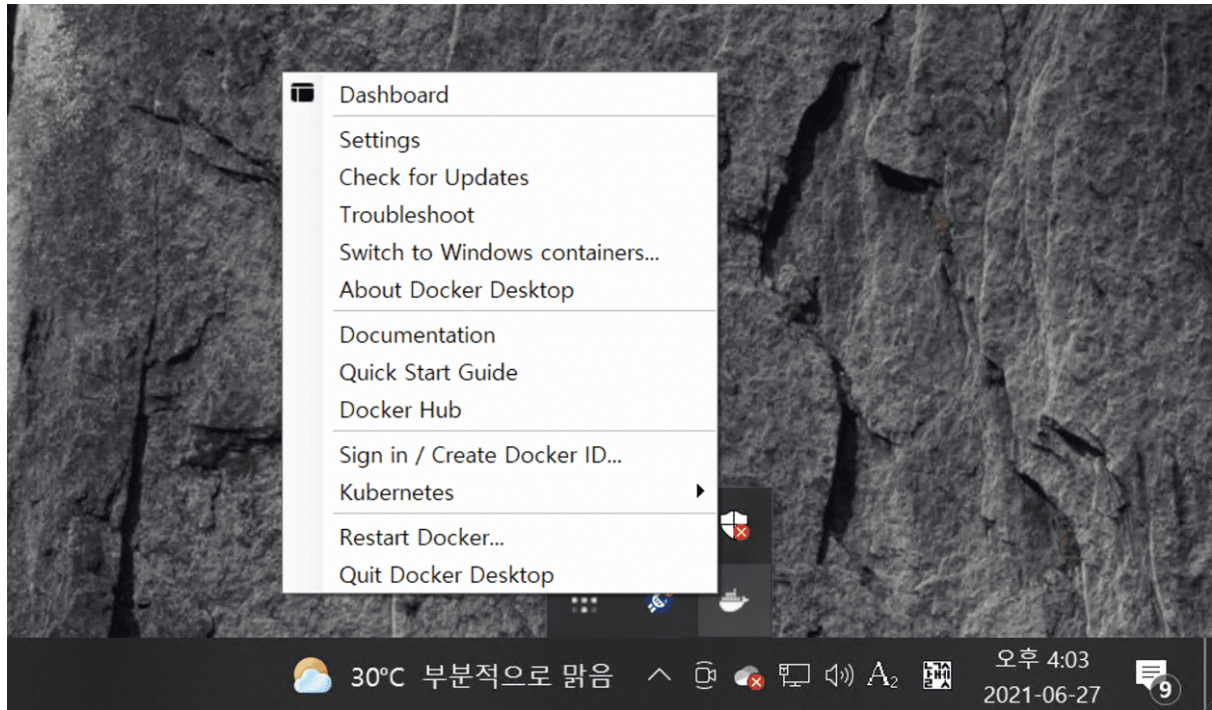
Docker Desktop을 실행합니다

7. 시스템에 WSL2가 활성화되어있다면 Docker는 기본적으로 WSL2를 백엔드로 Docker Engine을 실행합니다. 초기 셋업에는 몇 분 정도의 시간이 걸립니다. 성공적으로 Docker가 실행되면 Tutorial이 나타납니다.



Docker가 처음 실행되면 Tutorial 안내 페이지가 나타납니다

8. Docker Desktop은 시스템 트레이에 숨겨져있습니다. 숨겨진 아이콘을 활성화하고 고래 모양 아이콘에서 오른쪽 버튼을 누르면 상태를 Docker Desktop을 관리할 수 있습니다. 여기서 About Docker Desktop을 클릭합니다.



Windows 시스템 트레이의 Docker 메뉴

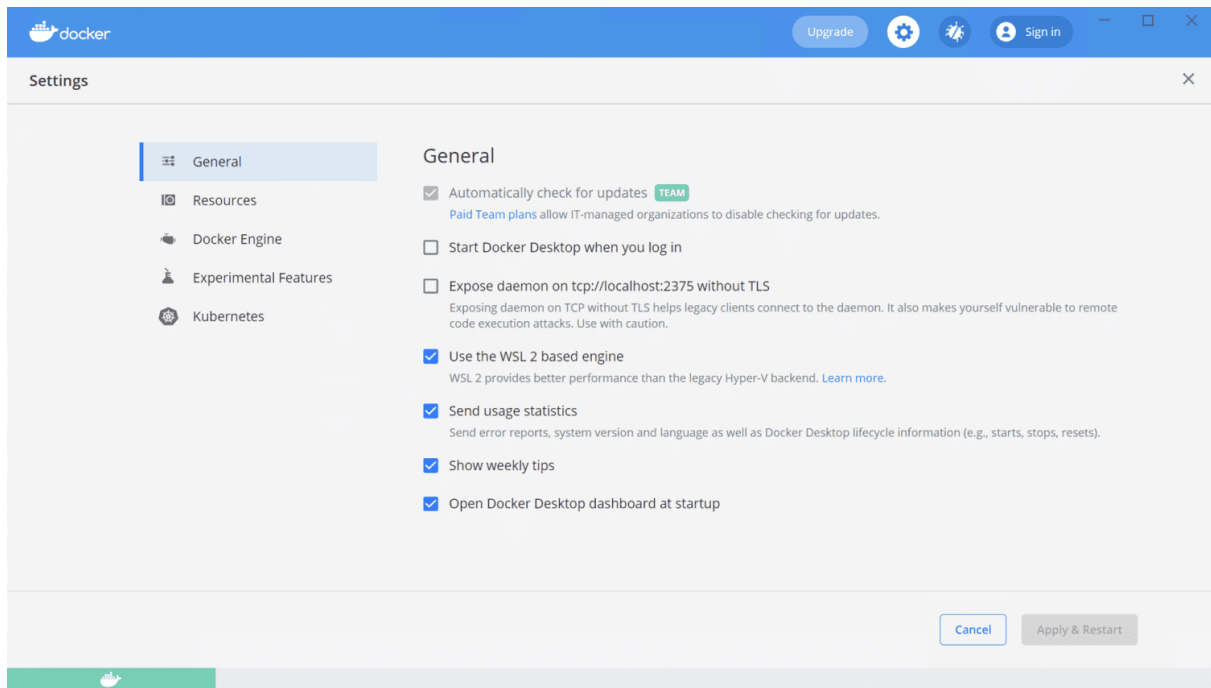
현재 설치된 Docker Desktop과 관련된 도구들의 버전을 여기서 확인할 수 있습니다.



About Docker Desktop에서 Docker 관련 도구의 버전을 확인할 수 있습니다

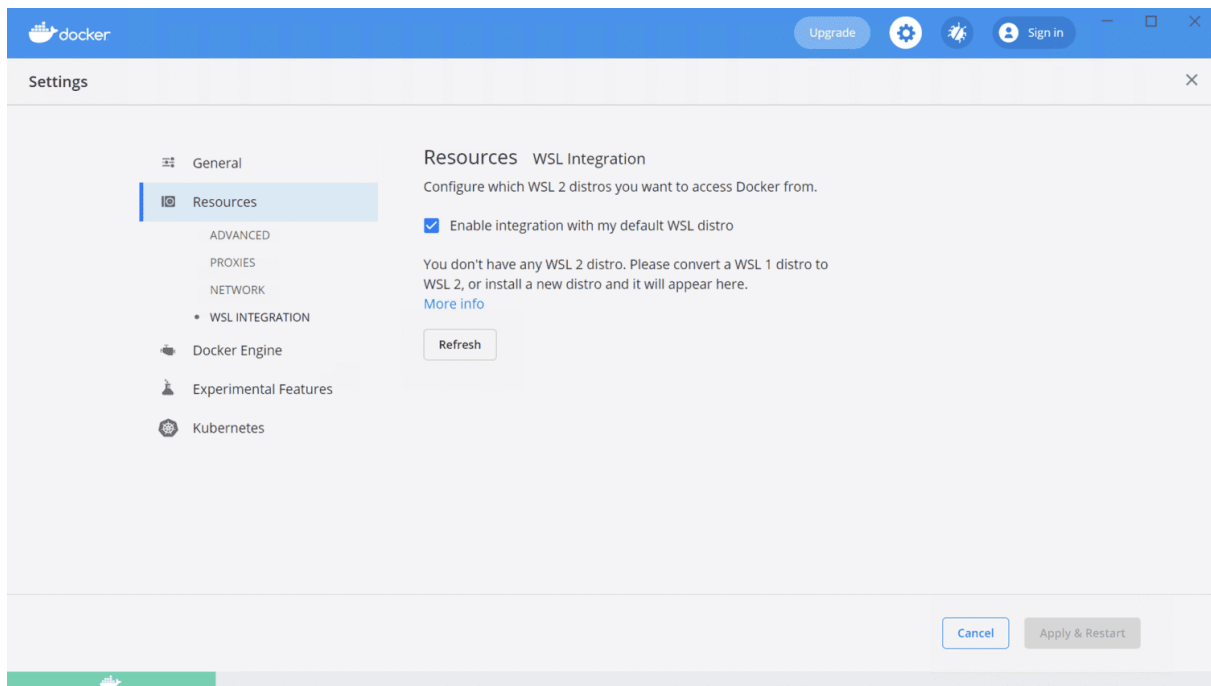
9. 다음으로 WSL2 설정이 잘되어있는지 확인하고 WSL 통합 설정을 진행하겠습니다.

Docker 아이콘에서 오른쪽 버튼을 눌러 Settings를 선택합니다. 먼저 General 설정에서 'Use the WSL 2 based engine'에 체크가 되어있는지 확인합니다. 미리 체크가 되어있을 텐데, 혹시 되어있지 않다면 체크하고 오른쪽 아래의 Apply & Restart 버튼을 클릭합니다.



WSL2 based engine 사용 여부가 체크되어있는지 확인합니다

10. 다음으로 왼쪽 사이드바에서 Resource > WSL Integration 메뉴로 이동합니다. 'Enable integration with my default WSL distro'에 체크되어있는지 확인합니다. 체크가 되어있지 않을 텐데 체크하고 오른쪽 아래의 Apply & Restart 버튼을 클릭해주면 도커 엔진이 재실행됩니다.



WSL2 통합 기능이 활성화 되어있는지 확인해봅니다

Docker가 재실행되면 WSL2 기반 Docker 설치 는 모두 완료되었습니다!

3.Docker 설치 확인 및 간단한 nginx 서버 예제 실행해보기(docker행)

Windows Terminal을 열어서 간단한 테스트를 해보겠습니다. PowerShell 탭을 하나 열고 `wsl` 명령어로 Docker 전용 머신이 실행중인 것을 확인할 수 있습니다.

```
$ wsl -l -v
```

NAME	STATE	VERSION
* docker-desktop	Running	2
docker-desktop-data	Running	2

wsl로 docker-desktop 리눅스에 명령어를 실행해볼 수 있습니다. docker-desktop은 BusyBox 기반의 경량 리눅스인 것을 확인해볼 수 있습니다.

```
$ wsl -d docker-desktop busybox
BusyBox v1.29.3 (2019-01-24 07:45:07 UTC) multi-call binary
y.
...
```

`docker version` 명령으로 Docker 서버와 클라이언트 정보를 확인해봅니다.

```
$ docker version
Client:
Cloud integration: 1.0.17
Version:          20.10.7
API version:      1.41
...

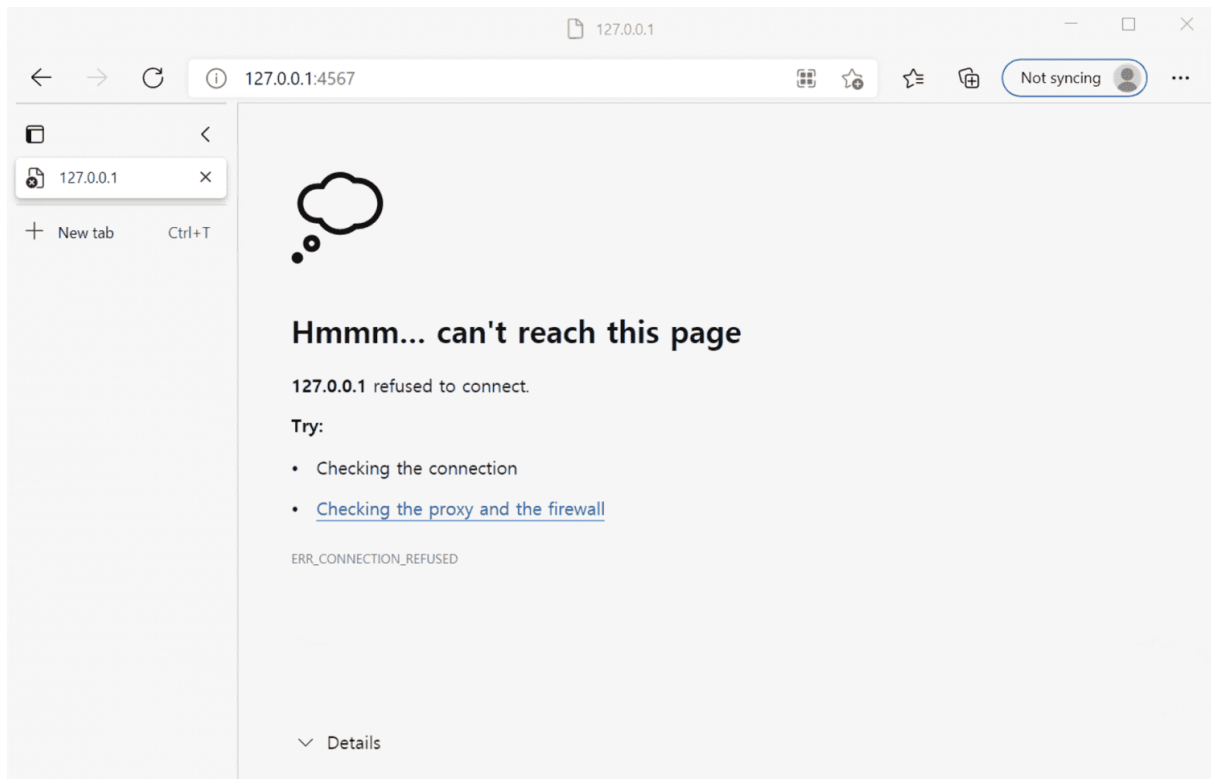
Server: Docker Engine - Community
Engine:
Version:          20.10.7
API version:      1.41 (minimum version 1.12)
...
```

`docker ps` 로 실행중인 컨테이너를 확인해봅니다.

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORT
S	NAMES				

아직 아무것도 실행중이지 않은 것을 확인할 수 있습니다. 여기서부터는 nginx 이미지로 간단한 서버 테스트를 해보겠습니다. 먼저 웹 브라우저를 열어 127.0.0.1:4567에 접속해봅니다. 다음과 같이 사이트에 접속을 할 수 없는 상태인 것을 확인합니다.



4567 포트로 접속해도 아무것도 나타나지 않습니다

`docker run` 명령어로 nginx 이미지 기반 컨테이너를 하나 실행해봅니다.

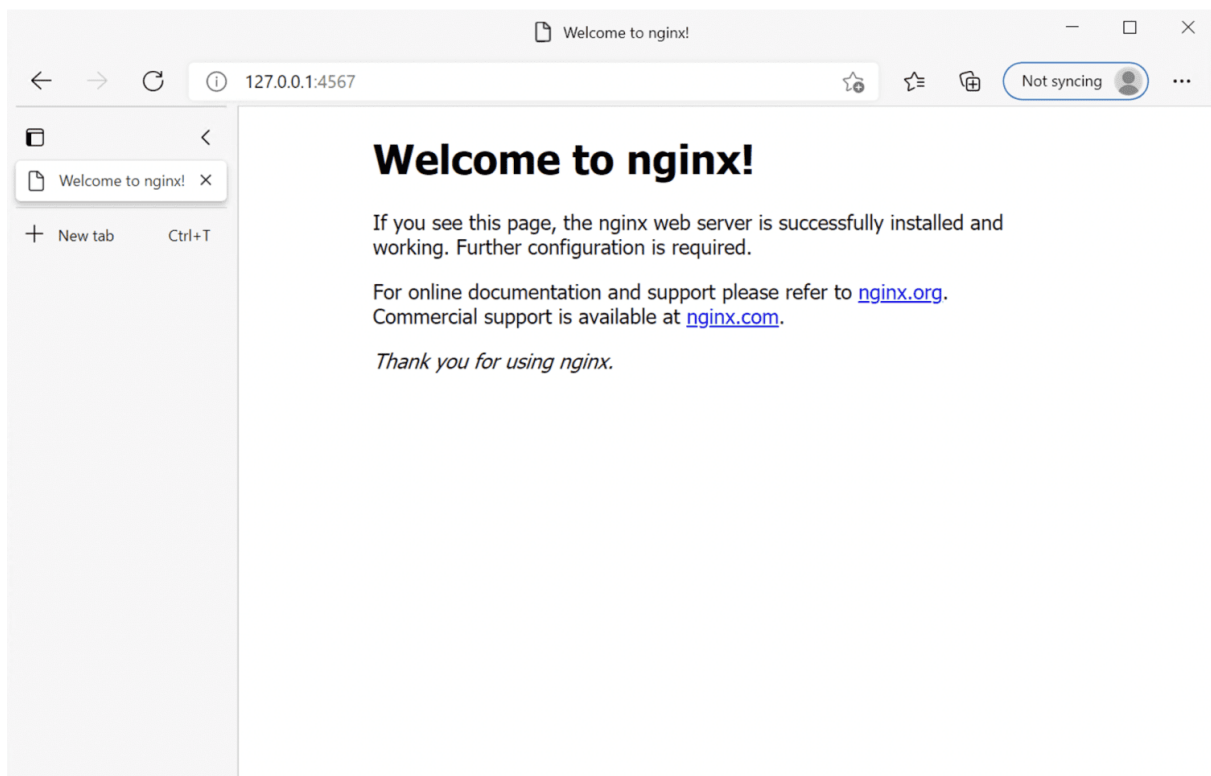
```
$ docker run -p 4567:80 -d nginx:latest
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
b4d181a07f80: Pull complete
edb81c9bc1f5: Pull complete
b21fed559b9f: Pull complete
03e6a2452751: Pull complete
b82f7f888feb: Pull complete
```

```
5430e98eba64: Pull complete
Digest: sha256:47ae43cdfc7064d28800bc42e79a429540c7c80168e8
c8952778c0d5af1c09db
Status: Downloaded newer image for nginx:latest
5909b49c4a0e677fae0146846bdf4fecae1ef869ece4cc46499f58d00f31
1e3d1
```

Docker에서는 이미지를 자동으로 다운로드 받고 실행해줍니다. `docker ps` 로 실행한 컨테이너를 확인합니다.

```
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED
STATUS        PORTS          NAMES
5909b49c4a0e   nginx:latest   "/docker-entrypoint...." 1 se
cond ago      Up Less than a second  0.0.0.0:4567->80/tcp,
:::4567->80/tcp   musing_allen
```

다시 웹 브라우저에서 127.0.0.1:4567에 접속해보면, 이제 'Welcome to nginx!' 메시지가 나타납니다.



4567 포트로 nginx 서버가 실행중인 것을 확인할 수 있습니다

사용하지 않는 컨테이너는 `docker rm` 명령어로 삭제해줍니다. `5909b49c4a0e` 는 `docker ps` 에서 확인할 수 있는 컨테이너 ID입니다.

```
$ docker rm -f 5909b49c4a0e
```

또 다른 실행 확인 방법

WindowsTerminl 에서

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