

가상화

가상화는 하나의 서버를 여러 개로 나눠서 사용하게 해줍니다.

1.가상화와 Docker의 개요

가상화는 물리적 컴퓨터를 여러 개의 가상 컴퓨터로 나누는 마술 같은 기술입니다.

이는 하나의 장비를 여러 대의 PC처럼 운영할 수 있게 해줍니다.

DOCKER



가상화란 무엇인가?

가상화의 개념

가상화는 한 컴퓨터를 여러 대의 가상의 작은 컴퓨터로 나누는 기술입니다. 마치 큰 집의 방들을 나누어 여러 사람이 각자 사용할 수 있게 만드는 것과 같습니다.

가상화의 장점

컴퓨터를 여러 대 사지 않아도 다양한 환경을 사용할 수 있습니다. 또한, 한 작업이 문제가 생겨도 다른 작업에 영향을 주지 않습니다.

실생활 비유

가상화는 큰 집의 방들을 나누는 것과 같습니다. 각 방이 다른 역할을 맡듯이, 가상화로 만들어진 작은 컴퓨터들도 서로 다른 일을 할 수 있습니다.

Docker의 특징 : 가상화

Docker는 컴퓨팅 환경에서 **가상화 기술**의 한 형태로, 컴퓨터 내에 독립적인 소규모 환경(컨테이너)을 구축합니다.

이 환경은 다양한 운영체제나 프로그램이 독립적으로 실행될 수 있는 공간을 제공합니다. 즉, 한 대의 컴퓨터에서 다양한 운영체제를 동시에 사용하는 것과 유사합니다. 예를 들어, Windows 컴퓨터 사용 시 Docker를 활용하여 Ubuntu Linux를 구현할 수 있습니다.

Docker 가상의 식당

Docker 식당

Docker를 식당에 비유하면 쉽게 이해할 수 있습니다. Docker 식당에는 필요한 모든 도구와 재료가 정리되어 있어 어디서나 같은 요리(프로그램)를 만들 수 있습니다.

이와 같이, Docker는 하나의 '가상식당(Container)'에 프로그램 실행에 필요한 모든 것을 포함합니다. 따라서, 다른 환경에서도 똑같이 프로그램을 실행할 수 있습니다. 또한, 여러 식당을 운영하는 것과 같은 가상화 기능도 제공합니다.

Docker의 장점

보통 프로그램은 특정한 운영체제에서만 작동됩니다. 만약 우리가 만든 프로그램이 Linux에서만 작동한다면, 이를 Windows에서 실행하려면 새로운 도구를 설치해야 합니다.

Docker는 이 프로그램과 필요한 환경을 하나의 가상 식당(Container)으로 만들어줍니다. 이 가상 식당은 어디서나 사용할 수 있어 예를 들어 본 식당 (Windows 운영체제) 에서도 가상식당 (ubuntu 컨테이너)을 실행할 수 있도록 해서 마치 Windows 컴퓨터와 Ubuntu 컴퓨터 두대의 컴퓨터가 있는것 처럼 사용 할 수 있습니다.

2. 프로그램 개발과 Docker 사용

Docker로 개발 환경을 표준화하여, 다양한 환경에서 프로그램을 실행할 수 있습니다.

이로 인해 크로스 플랫폼 호환성을 얻고, 개발 및 배포가 효율적입니다.

개발 환경과 실행 환경의 차이

개발자가 프로그램을 만들고 테스트하는 환경은 보통 Windows 같은 운영체제를 사용합니다.

하지만 실제로 프로그램이 실행되는 AWS는 Ubuntu 같은 리눅스 환경을 사용하는 경우가 많습니다.

이 두 환경은 다음과 같은 이유로 서로 다릅니다

운영체제의 차이

개발 환경인 Windows와 실행하는 환경 Ubuntu는 프로그램이 동작하는 방식이 다릅니다.

환경의 실제 예

개발 환경 Windows에서 잘 작동하는 프로그램이 실행 환경 Ubuntu에서는 오류 가능.

파일 경로의 차이

개발환경 Windows는 파일 경로를 C:\폴더\파일처럼 사용하지만, 실행 환경 Ubuntu는 /폴더/파일처럼 다르게 사용합니다.

필요한 라이브러리와 설정

프로그램이 실행되려면 특정한 소프트웨어나 설정이 필요할 수 있는데, 개발 환경과 실행 환경이 다르면 설정이 맞지 않아 프로그램이 정상적으로 실행되지 않을 수 있습니다.

Docker 컨테이너를 이용한 환경 차이 극복

Docker는 개발 환경과 실행 환경을 일치시킴으로써 효율성을 극대화하는 도구입니다.

실행에 필요한 운영체제, 파일 경로, 소프트웨어 설정 등을 하나의 컨테이너(Container)로 구성하여, Windows에서 개발된 프로그램도 Ubuntu 환경에서 원활하게 실행될 수 있도록 설계되었습니다.

이 과정을 요리한 음식을 다른 식당에서 똑같이 재현해야 하는 상황에 비유해보겠습니다. 익숙한 식당에서 요리를 했지만, 다른 장소의 식당은 조리 도구와 환경이 전혀 다를 수 있습니다. 이런 차이로 인해, 동일한 요리를 완벽히 재현하기 어려워질 수 있습니다.

Docker의 역할은 이런 문제를 해결하기 위해, 하나의 이동 가능한 가상 식당(Container)을 준비하는 것입니다.

Docker 식당은 요리에 필요한 조리 도구, 재료, 그리고 환경이 모두 포함되어 있어, 어디로 옮기더라도 같은요리를 만들 수 있습니다.

이처럼 Docker는 프로그램이 다양한 환경에서도 안정적으로 작동하도록, 필요한 모든 환경을 식당처럼 구성해 제공합니다.

3.Windows 환경에서 Docker 설치

Docker 공식 웹사이트에서 Windows용 Docker 설치 프로그램을 다운로드하고 실행합니다.

3.1 가상화 가능 확인

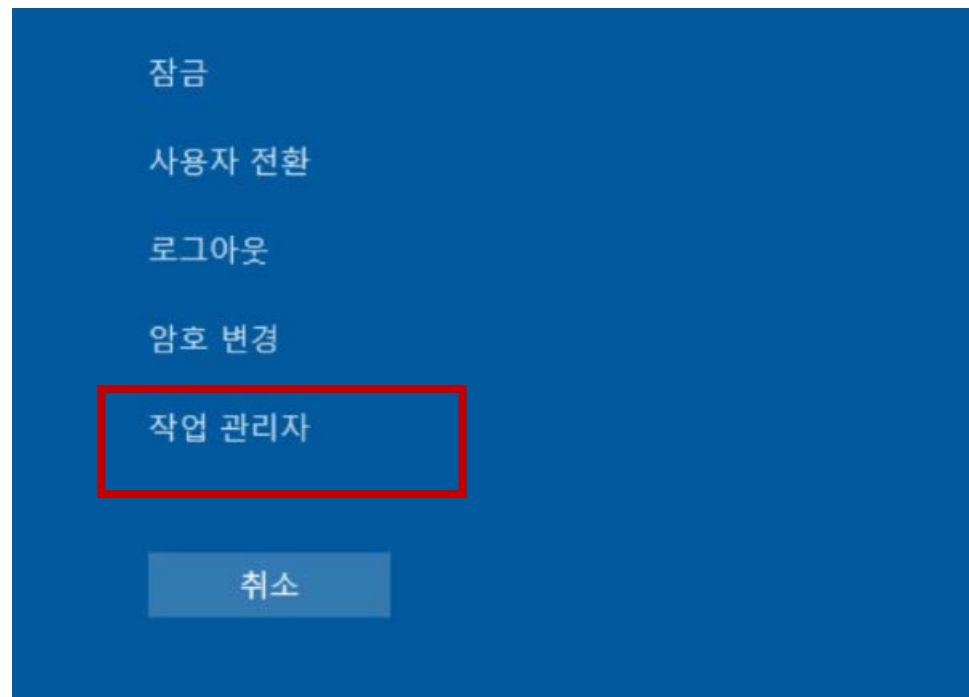
Windows 에서 가상화 가능 확인

가상화 가능 확인

가상화 기능을 사용 가능한지 확인

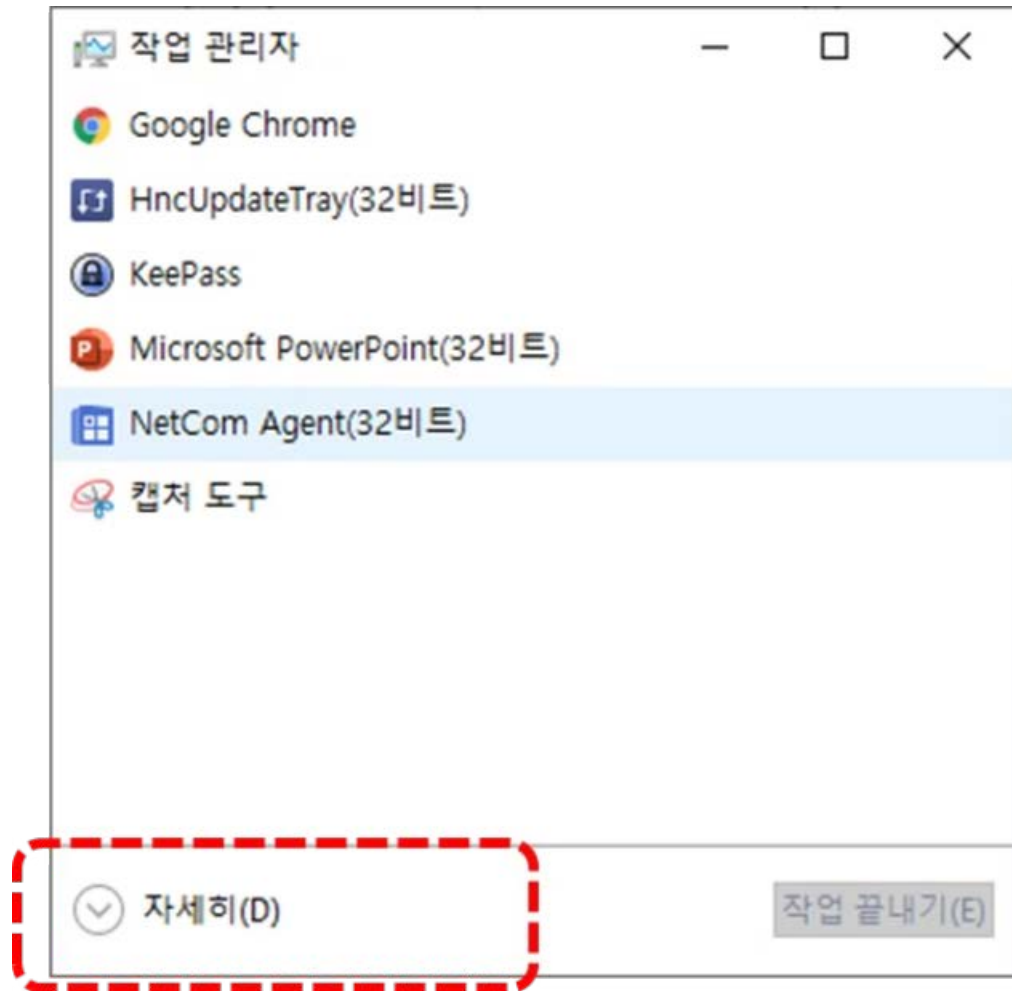
Ctrl Alt Del 키를 동시에 누름

파란 창에서 작업 관리자 선택



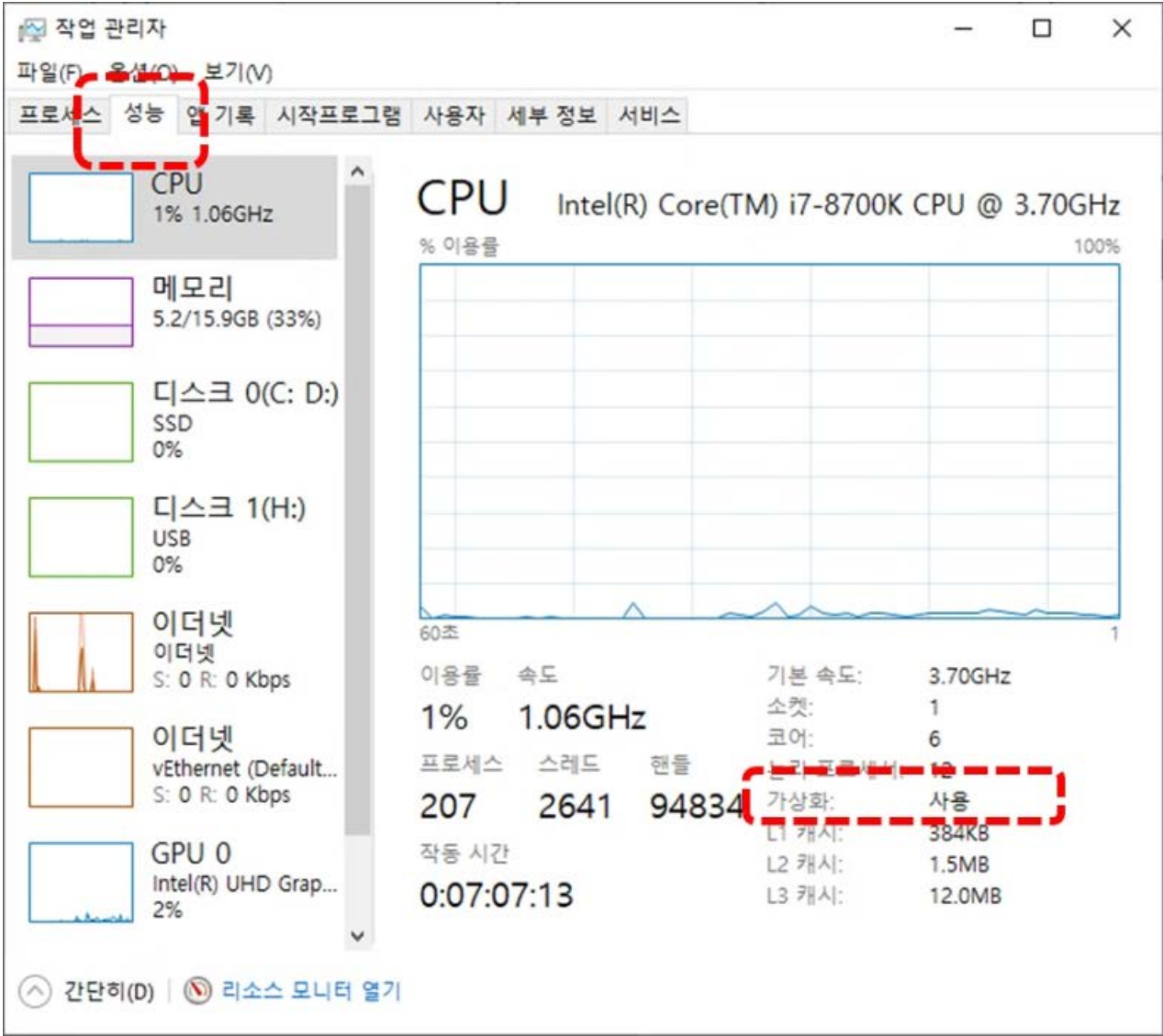
가상화 가능 확인

자세히 클릭



가상화 가능 확인

성능 탭 클릭 후 가상화가 사용으로 되어 있는지 확인

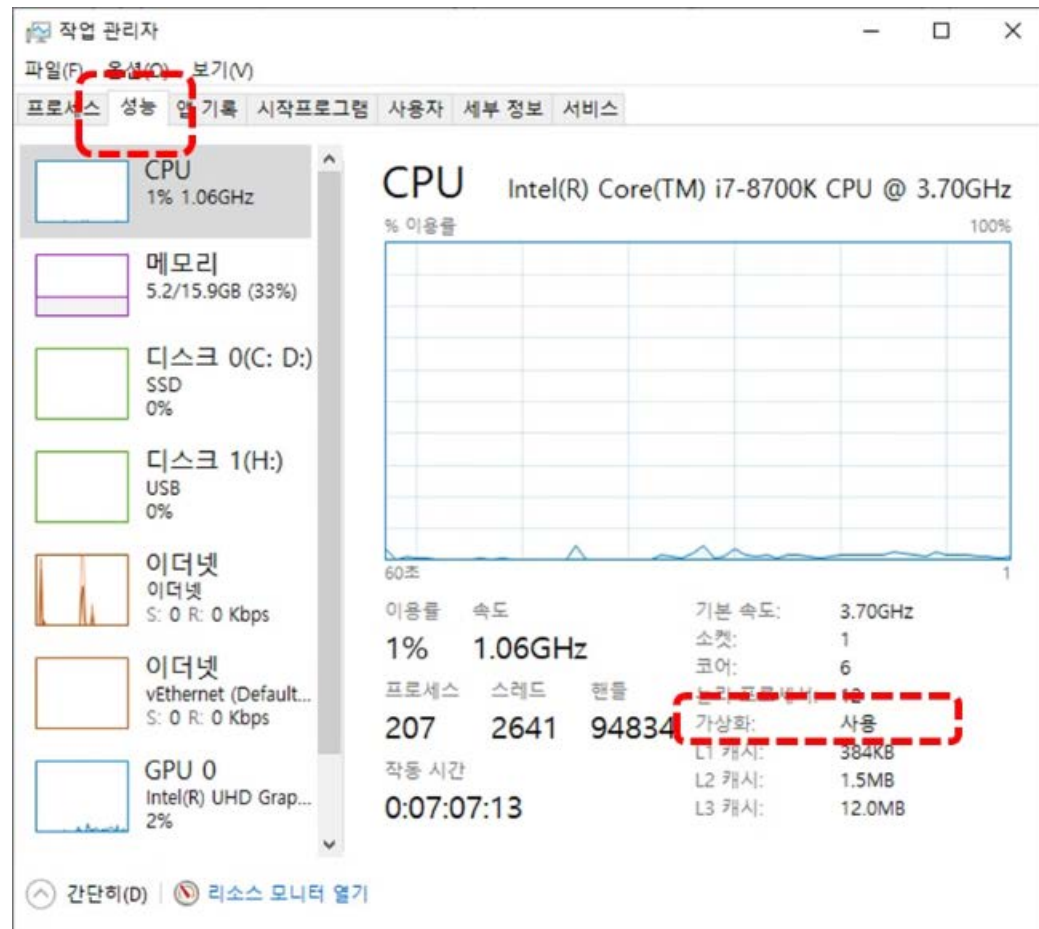


3.2 가상화가 가능하지 않을때 Bios 설정

가상화가 가능하지 않을때 Bios 설정

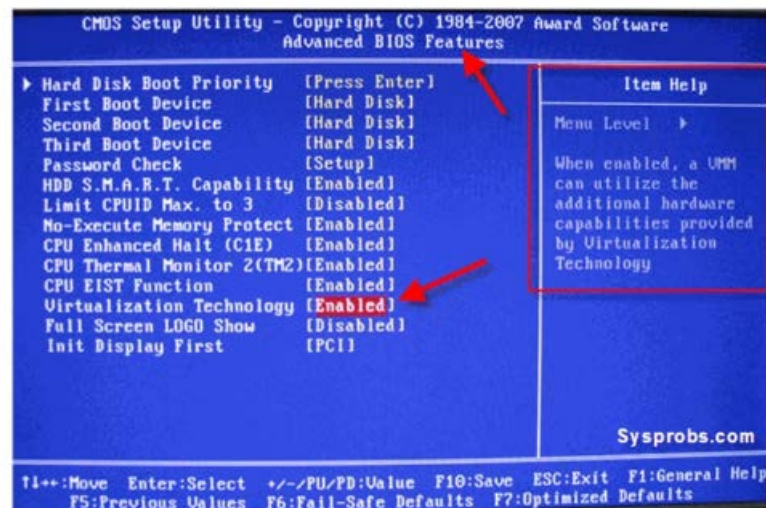
가상화가 가능하지 않을때 Bios 설정

가상화가 사용이 아닐때 Bios 설정을 해야함



가상화가 가능하지 않을때 Bios 설정 (HP 컴퓨터예시)

1. 컴퓨터 시스템 종료
2. 컴퓨터 전원 켜기
3. HP 컴퓨터 예시 시작되는 동안 반복적으로 Esc 키를 눌러 BIOS로 들어 감
4. BIOS Setup의 F10 키 입력 오른쪽 화살표 키를 눌러 시스템 구성 탭의 가상화 기술 선택로 이동 한 다음 엔터 키 입력



5. 사용을 선택하고 Enter 키 입력
6. F10 키를 누르고 예를 선택하고 Enter 키를 눌러 변경 사항을 저장하고 재부팅

가상화가 가능하지 않을때 Bios 설정 (레노버, 델, 아수스 예시)

레노버

<https://support.lenovo.com/in/en/solutions/ht500006-how-to-enable-virtualization-technology-on-lenovo-computers> 참조

델

1. BIOS에서 F2를 누릅니다.
2. 고급 탭에서 가상화 선택 후 엔터를 누릅니다.
3. 사용 선택 후 엔터, F10을 누르고 "예" 선택 엔터를 눌러 저장하고 재부팅합니다.

아수스

1. BIOS에서 F2를 사용.
2. 고급 탭에서 가상화 선택 후 엔터.
3. 사용 선택 후 F10을 누르고 "예" 선택 엔터를 눌러 저장하고 재부팅합니다.

가상화가 가능하지 않을때 Bios 설정 (에이서 예시)

1. 컴퓨터 시작 시 F2 키를 눌러 BIOS 설정으로 들어갑니다.
2. 오른쪽 화살표 키로 시스템 구성 탭의 가상화 기술로 이동한 후 엔터 키를 누릅니다.
3. '사용'을 선택하고 엔터 키를 누릅니다.
4. F10을 눌러 '예'를 선택한 뒤 엔터 키를 눌러 변경을 저장하고 재부팅합니다.

3.3 운영 체제 확인

윈도우즈 운영 체제 확인

운영 체제 확인

Windows 키" + "Pause 또는 Break " 라고 쓰여진 키를 누릅니다. 시스템 정보 창이 바로 열립니다.

윈도우 운영 체제 확인

Windows 사양

에디션	Windows 10 Pro
버전	22H2
설치 날짜	2020-09-15
OS 빌드	19045.5131
경험	Windows Feature Experience Pack 1000.19060.1000.0

복사

3.4 Windows Home 에디션 Hyper-V 설치

Windows Home 에디션에서는 Hyper-V를 설치해야 합니다.

Windows Home 에디션에서 Hyper-V 설치

Hyper-V 기능

Hyper-V는 윈도우에서 리눅스와 같은 가상 운영체제를 실행할 수 있는 기능입니다.

Home 에디션

Windows Home 에디션에는 Hyper-V가 기본으로 제공되지 않습니다. 아래의 링크를 참조해서 Hyper-V를 설치 해야 합니다.

<https://lastcard.tistory.com/141>

다른 에디션

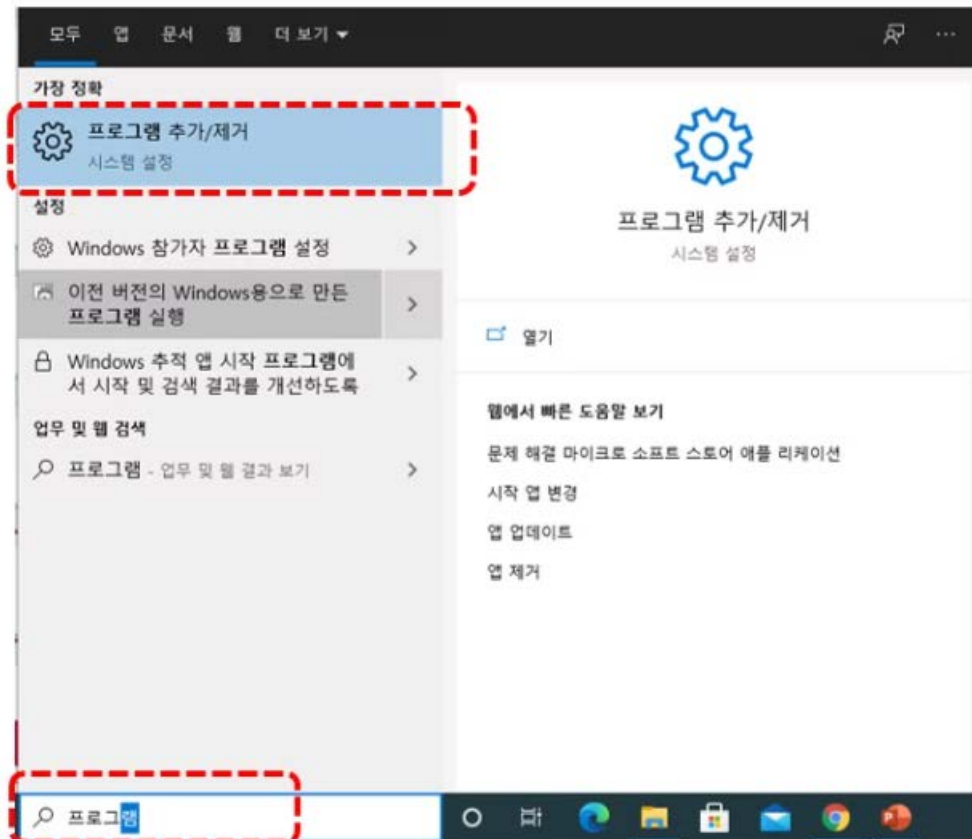
Windows Pro 및 Windows Enterprise 에디션에는 Hyper-V가 기본 설치되어 있어 추가 설치가 필요 없습니다.

3.5 Hyper-V 설정

Hyper-V 설치 후, 설정을 확인하고 필요 시 조정하십시오.

Hyper-V 설정

프로그램 추가 제거 검색 후 클릭



Hyper-V 설정

프로그램 및 기능 클릭

The screenshot shows the Windows Settings application window. On the left is a navigation pane with the following items: '설정' (Settings), '홈' (Home), '설정 검색' (Search settings), '앱' (Apps), '앱 및 기능' (Apps and features), '기본 앱' (Default apps), '오프라인 지도' (Offline maps), '웹 사이트용 앱' (Apps for your web browser), '비디오 재생' (Video playback), and '시작 프로그램' (Startup apps). The '앱 및 기능' (Apps and features) item is selected and highlighted with a blue bar. The main content area is titled '앱 및 기능' (Apps and features) and contains the following text: '앱을 가져올 위치 선택' (Where to get apps), 'Microsoft Store에서 받은 앱만이 안전할 수 있습니다.' (Only apps from the Microsoft Store can be safe.), a dropdown menu set to '제한 없음' (No restrictions), and a section titled '앱 및 기능' (Apps and features) with sub-sections '선택적 기능' (Optional features) and '앱 실행 별칭' (App execution aliases). Below these is the text '드라이브별로 검색, 정렬 및 필터링합니다. 앱을 제거하거나 이동하려면 목록에서 선택하세요.' (Search, sort, and filter by drive. Select from the list to remove or move an app.) and a search box labeled '이 목록 검색' (Search this list). On the right side of the main content area, there is a '관련 설정' (Related settings) section with a red dashed box around the link '프로그램 및 기능' (Programs and Features). Below this is a '웹 도움말' (Web help) section with links to '문제 해결 마이크로 소프트 스토어 애플리케이션' (Troubleshoot Microsoft Store app), '시작 앱 변경' (Change start-up app), '앱 업데이트' (Update app), and '앱 제거' (Remove app). At the bottom right, there are links for '도움말 보기' (View help) and '피드백 제공' (Provide feedback).

설정

홈

설정 검색

앱

앱 및 기능

기본 앱

오프라인 지도

웹 사이트용 앱

비디오 재생

시작 프로그램

앱 및 기능

앱을 가져올 위치 선택

Microsoft Store에서 받은 앱만이 안전할 수 있습니다.

제한 없음

앱 및 기능

선택적 기능

앱 실행 별칭

드라이브별로 검색, 정렬 및 필터링합니다. 앱을 제거하거나 이동하려면 목록에서 선택하세요.

이 목록 검색

관련 설정

프로그램 및 기능

웹 도움말

문제 해결 마이크로 소프트 스토어 애플리케이션

시작 앱 변경

앱 업데이트

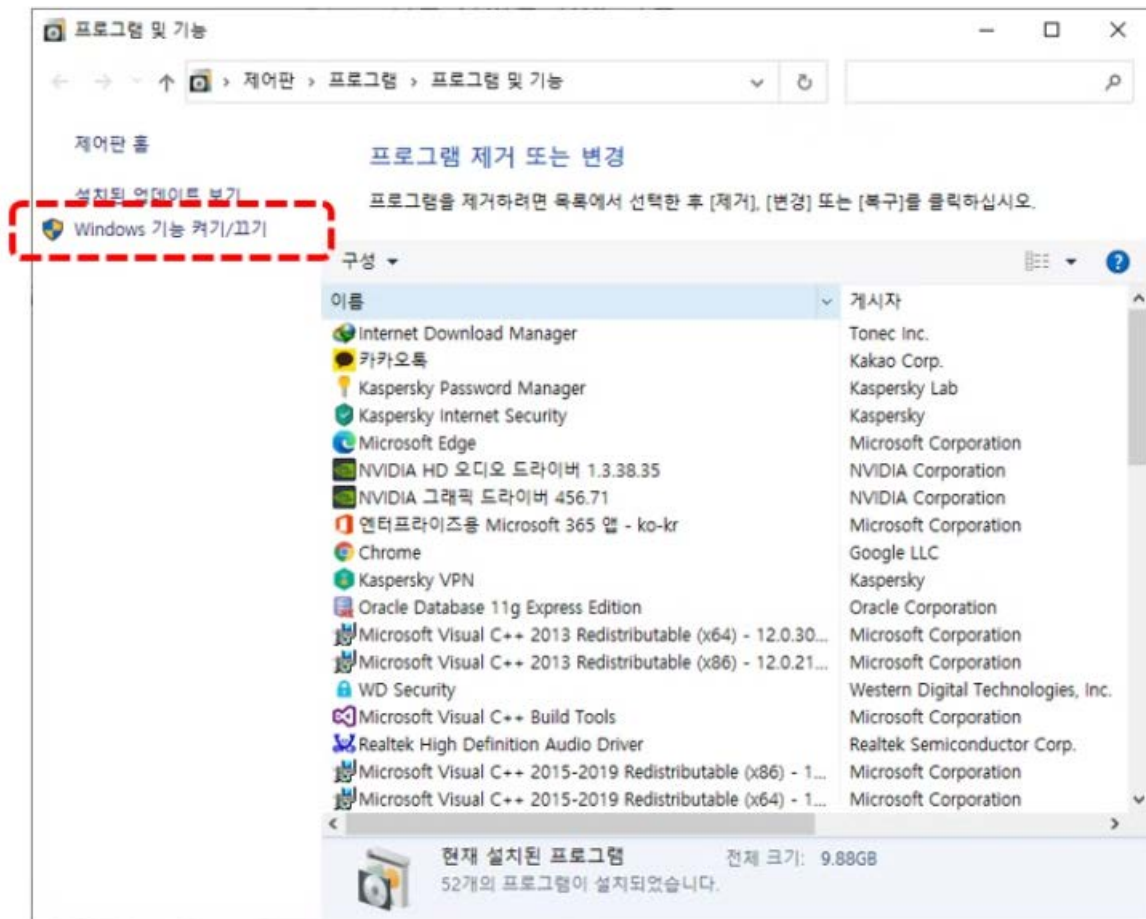
앱 제거

도움말 보기

피드백 제공

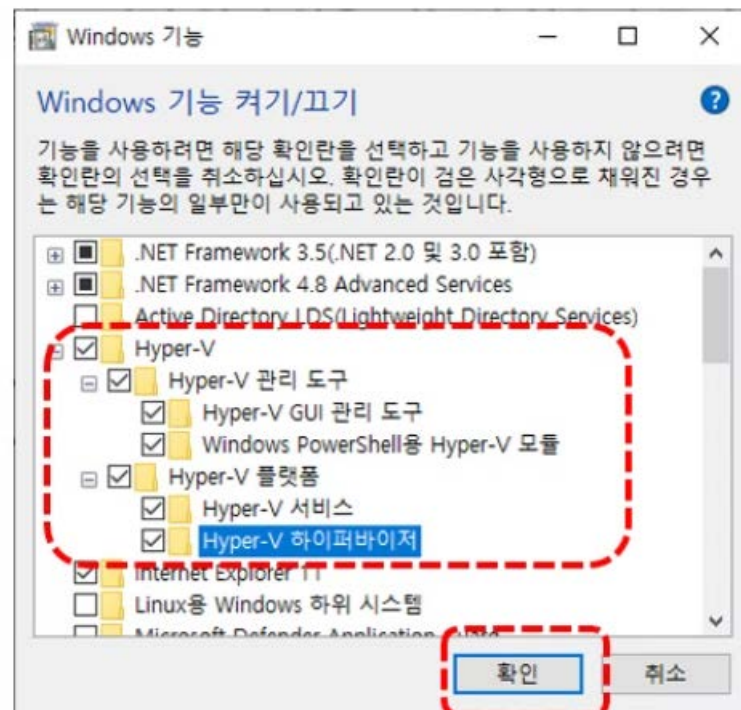
Hyper-V 설정

Windows 기능 켜기/ 끄기 클릭



Hyper-V 설정

1. 모든 Hyper-V 항목이 체크되었는지 확인
 1. 체크되지 않은 기능이 있으면 모두 선택
2. 확인 클릭
3. 컴퓨터 재부팅



3.6 WSL 커널 업데이트

WSL2 설치를 위한 커널 업데이트가 필요합니다.

WSL 커널 업데이트

1. <https://learn.microsoft.com/ko-kr/windows/wsl/install-manual> 페이지 접속
2. 화면 중간에 링크를 클릭해서 파일 다운로드
3. 다운로드 받은 파일을 기본 옵션으로 설치
4. 컴퓨터 재시작

이전 버전 WSL의 수동 설치 단계 x +

learn.microsoft.com/ko-kr/windows/wsl/install-manual

머신을 다시 시작하여 WSL 설치를 완료하고 WSL 2로 업데이트합니다.

4단계 - Linux 커널 업데이트 패키지 다운로드

1. 최신 패키지를 다운로드합니다

- [x64 머신용 최신 WSL2 Linux 커널 업데이트 패키지](#)

① 참고

ARM64 머신을 사용하는 경우 [ARM64 패키지](#)를 대신 다운로드하세요. 사용하고 있는 머신의 종류를 잘 모르는 경우 명령 프롬프트 또는 PowerShell을 열고 `systeminfo | find "System Type"`을 입력합니다. 주의: 비 영어 Windows 버전에서는 "시스템 유형" 문자열을 변환하여 검색 텍스트를 수정해야 할 수 있습니다. find 명령에 대한 따옴표는 이스케이프해야 할 수도 있습니다. 예를 들어 독일어 `systeminfo | find '"Systemtyp"'`입니다.

3.7 Docker 설치

WSL 커널 업데이트 후에는 Docker를 설치해야합니다.

Docker 설치

<https://hub.docker.com/editions/community/docker-ce-desktop-windows/> 다운로드

[Home](#) / [Manuals](#) / [Docker Desktop](#) / [Setup](#) / [Install](#) / Windows

Install Docker Desktop on Windows

Docker Desktop terms

Commercial use of Docker Desktop in larger enterprises (more than 250 employees OR more than \$10 million USD in annual revenue) requires a [paid subscription](#) [↗](#).

This page contains the download URL, information about system requirements, and instructions on how to install Docker Desktop for Windows.

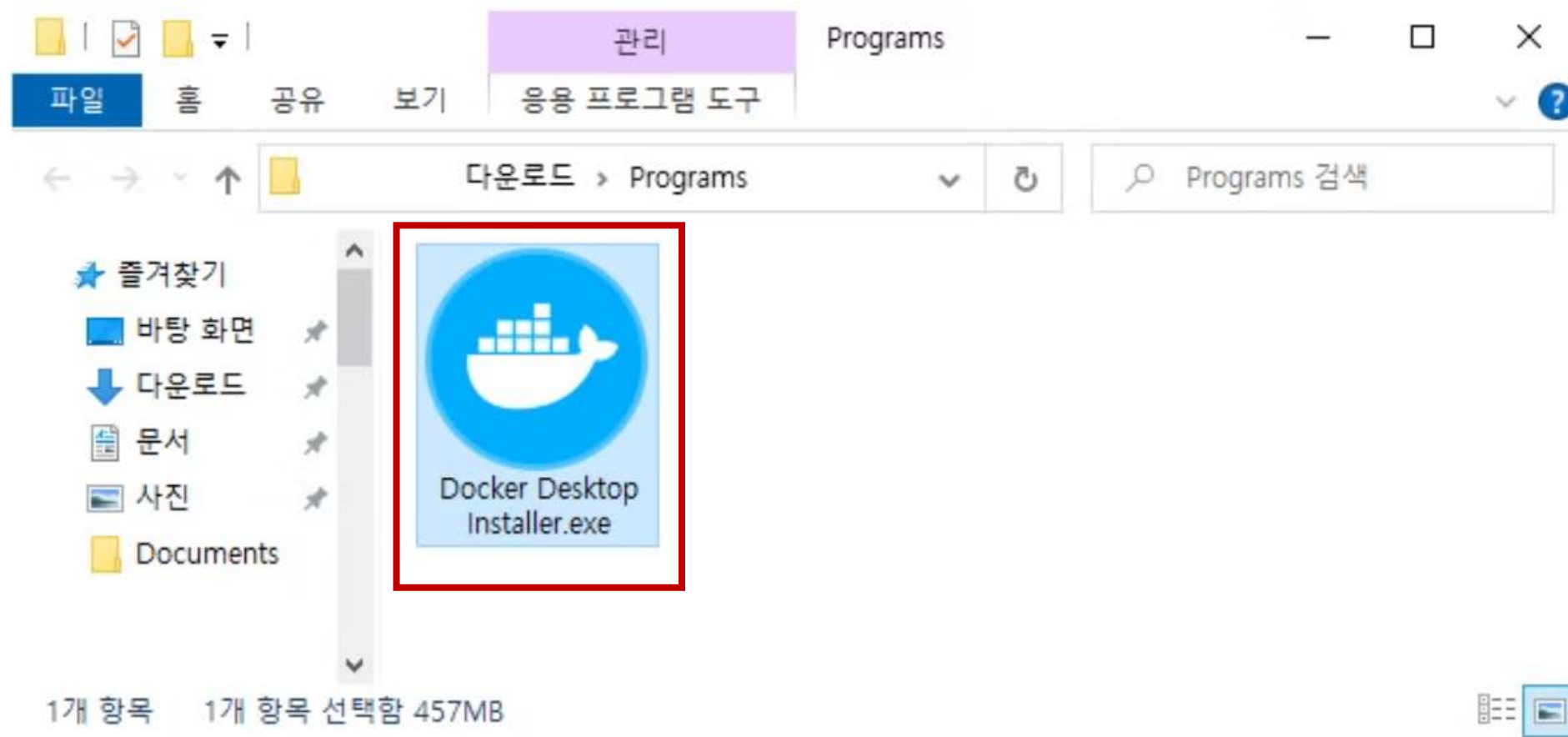
Docker Desktop for Windows - x86_64

Docker Desktop for Windows - Arm (Beta)

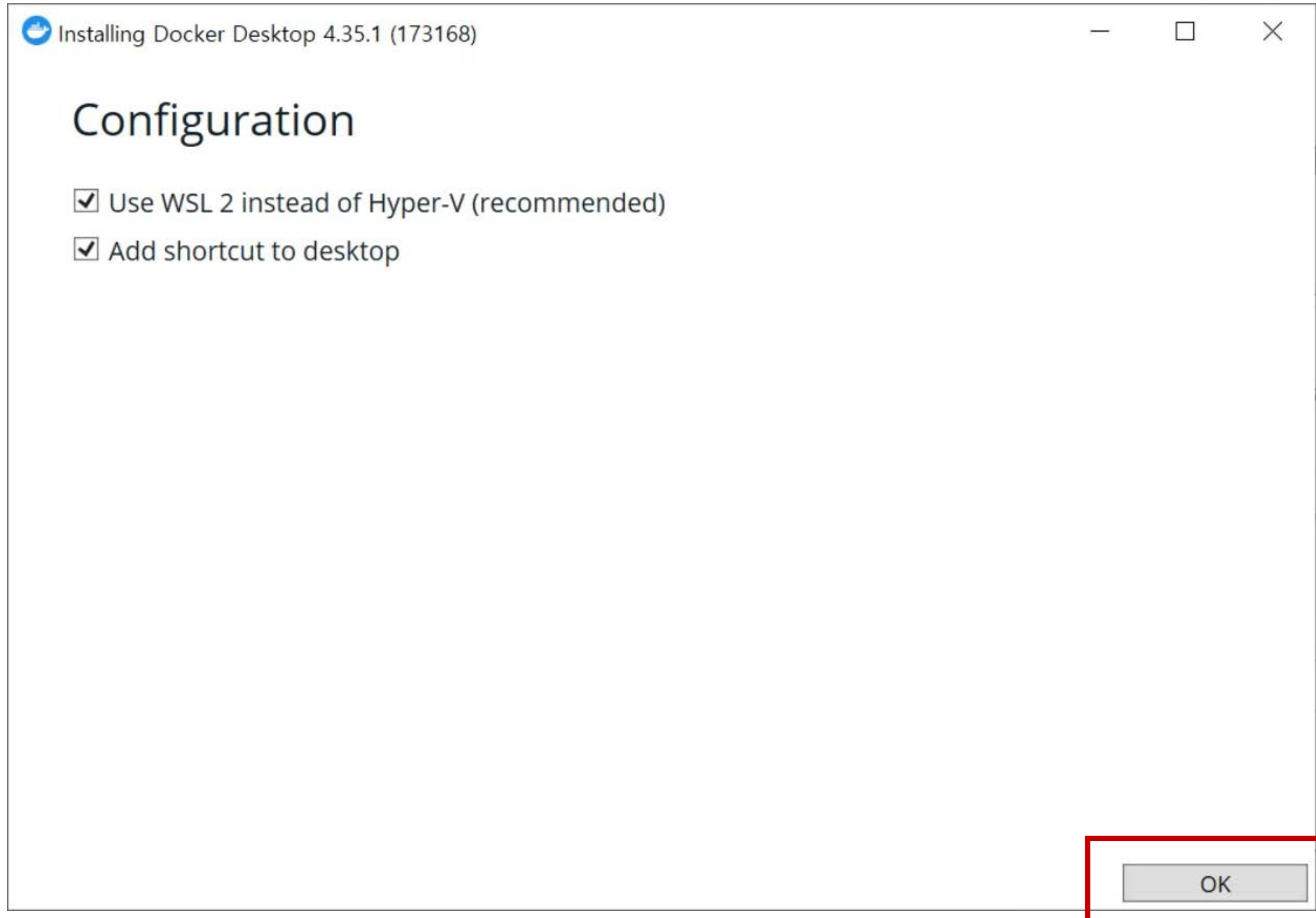
For checksums, see [Release notes](#)

Docker 설치

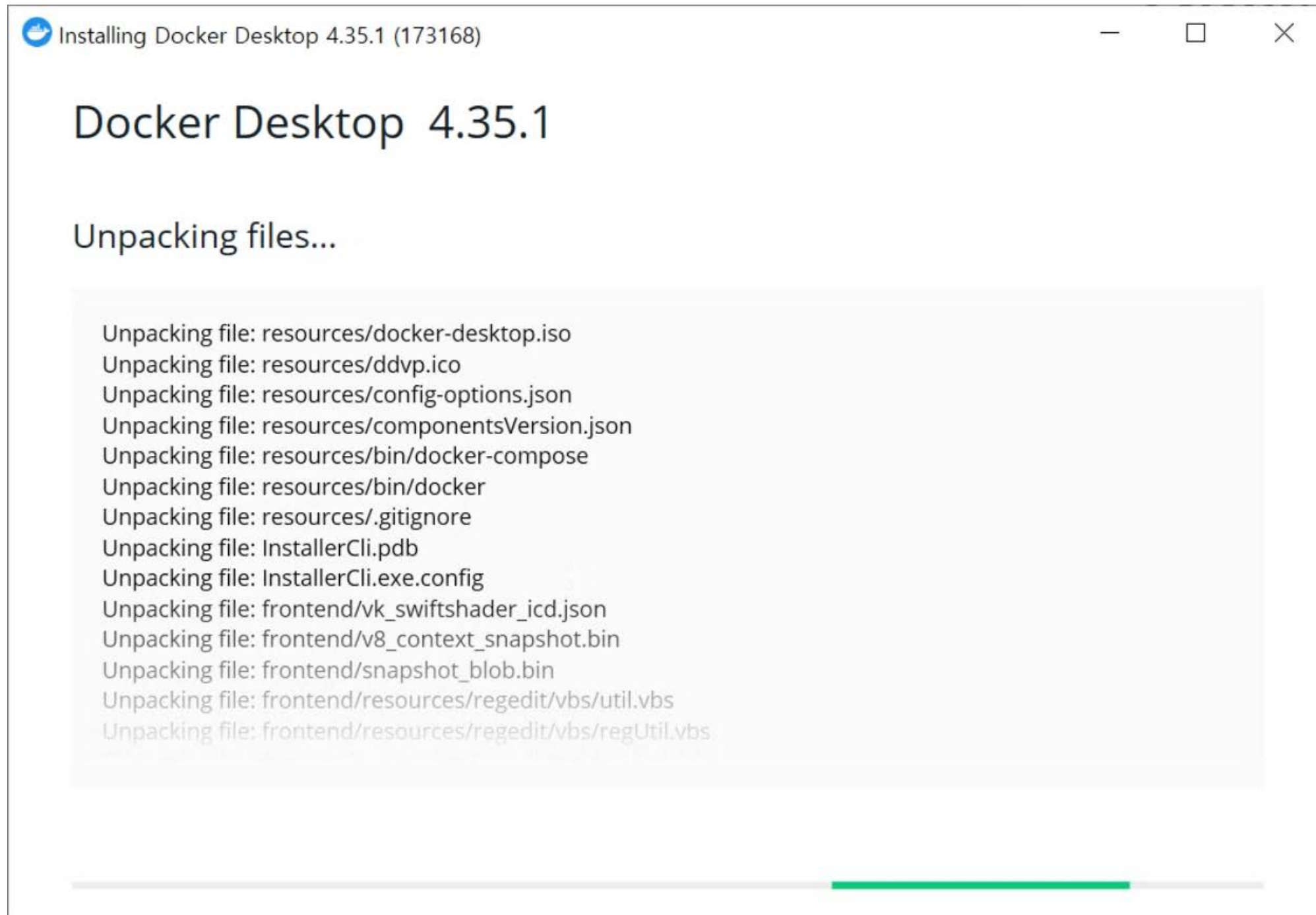
다운로드 받은 파일을 더블 클릭



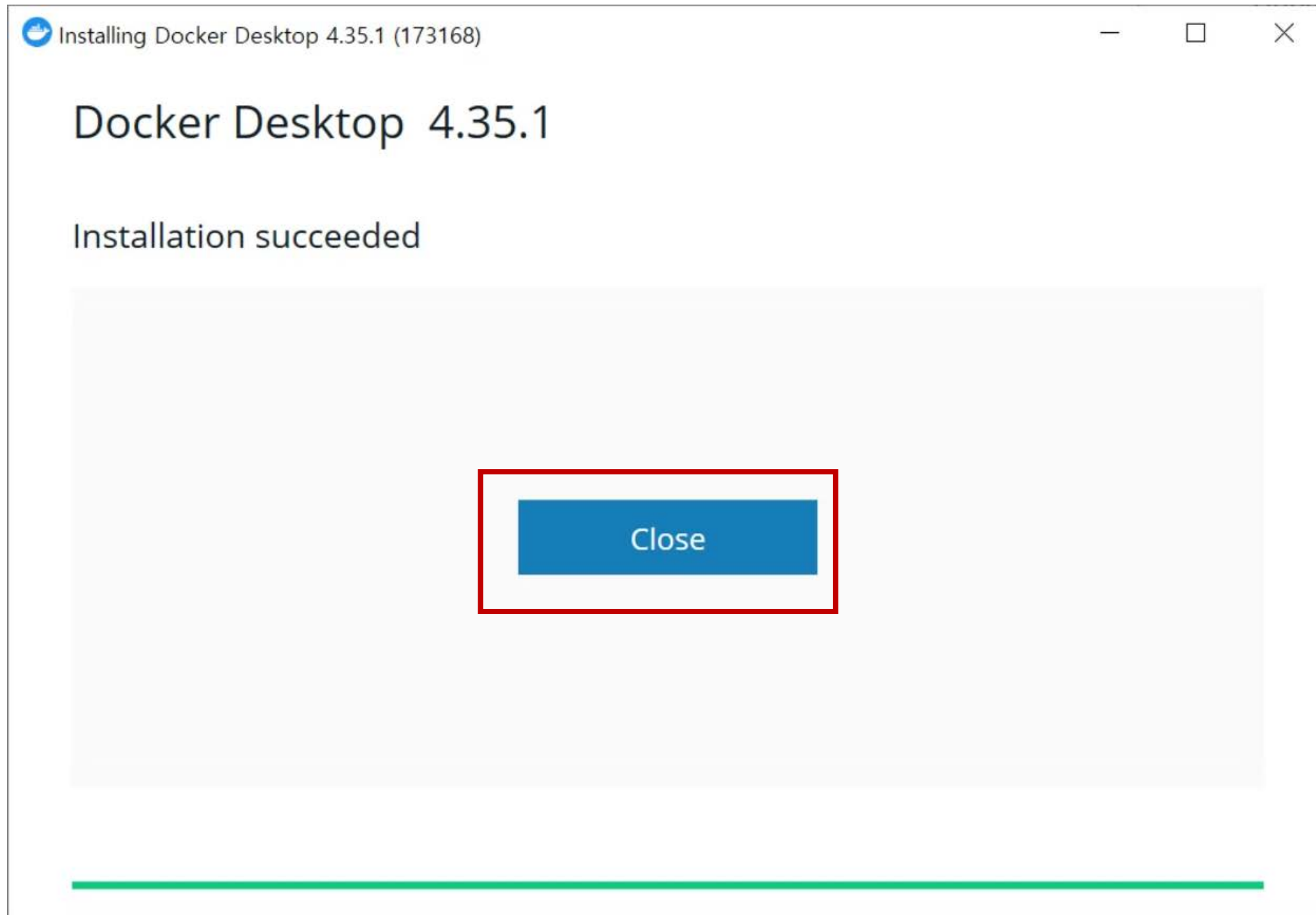
Docker 설치



Docker 설치



Docker 설치



3.8 Docker 실행

설치가 완료된 후, Docker 아이콘을 클릭하여 실행하세요.

Docker 실행

바탕화면에서 Docker Desktop 아이콘 더블클릭



Docker 실행



Docker Subscription Service Agreement

By selecting accept, you agree to the [Subscription Service Agreement](#), the [Docker Data Processing Agreement](#), and the [Data Privacy Policy](#).

Note: Docker Desktop is free for small businesses (fewer than 250 employees AND less than \$10 million in annual revenue), personal use, education, and non-commercial open source projects. Otherwise, it requires a paid subscription for professional use. Paid subscriptions are also required for government entities. [Read the FAQ to learn more.](#)

[View Full Terms](#) 

[Accept](#) [Close](#)

Docker 실행

[Skip](#)

Welcome to Docker

Work


Personal


Using Docker for work?
We recommend signing in with your work email address.


Work email address

Continue


Or






Create an account 

Docker 실행

Skip



Welcome Survey

Step 1 of 2

What's your role?

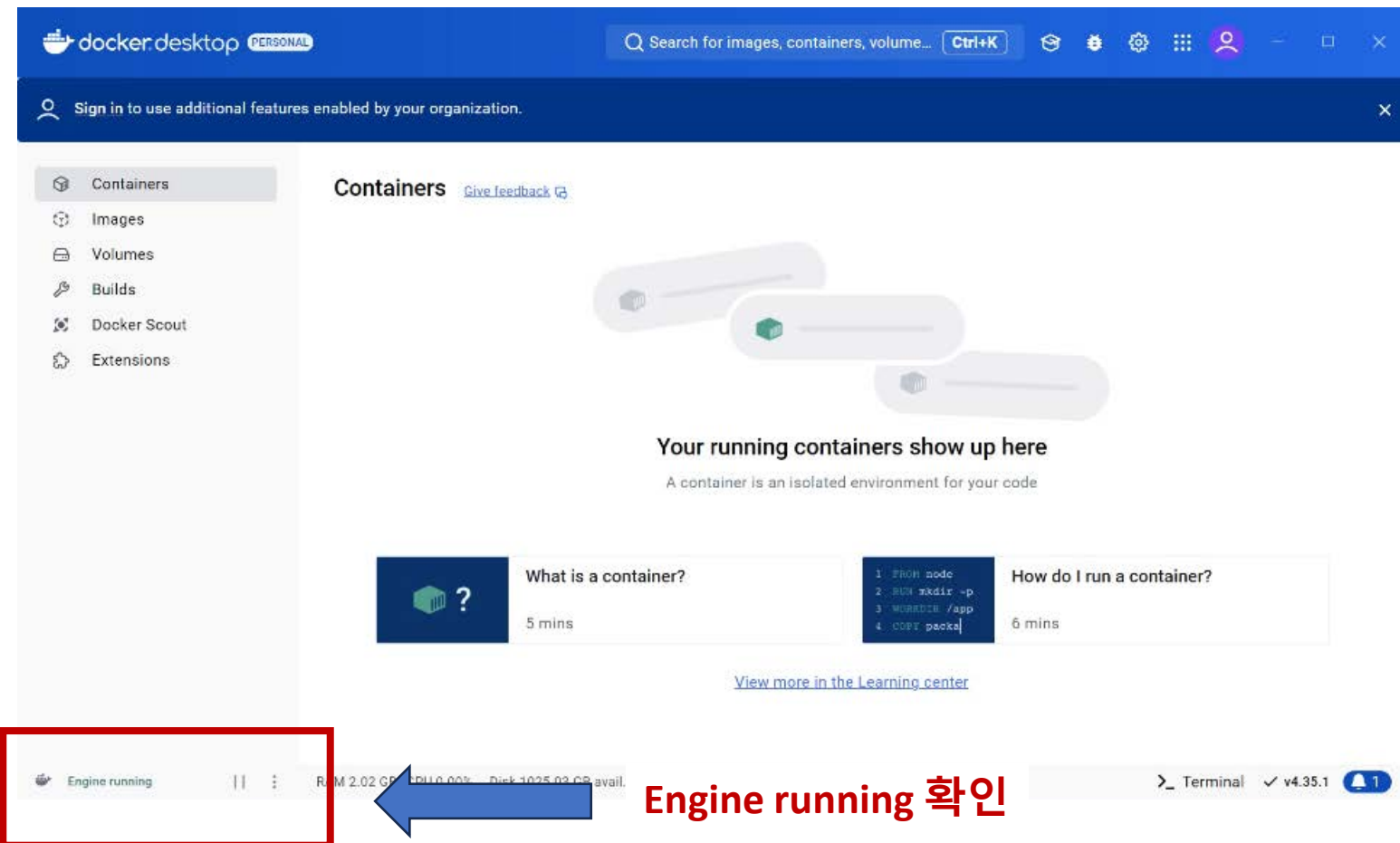
<input type="radio"/> Full-stack developer	<input type="radio"/> Front-end developer
<input type="radio"/> Back-end developer	<input type="radio"/> Site reliability engineer
<input type="radio"/> Platform engineer	<input type="radio"/> DevOps specialist
<input type="radio"/> Infrastructure manager	<input type="radio"/> System administrator
<input type="radio"/> Security engineer	<input type="radio"/> Data scientist
<input type="radio"/> Product manager	<input type="radio"/> Teacher
<input type="radio"/> Student	<input type="radio"/> Other

Docker 실행

Engine running 메시지를 확인하세요.

이 메시지가 없으면 10분 정도 기다리세요.

10분 후에도 나오지 않으면 설치가 실패한 것입니다.

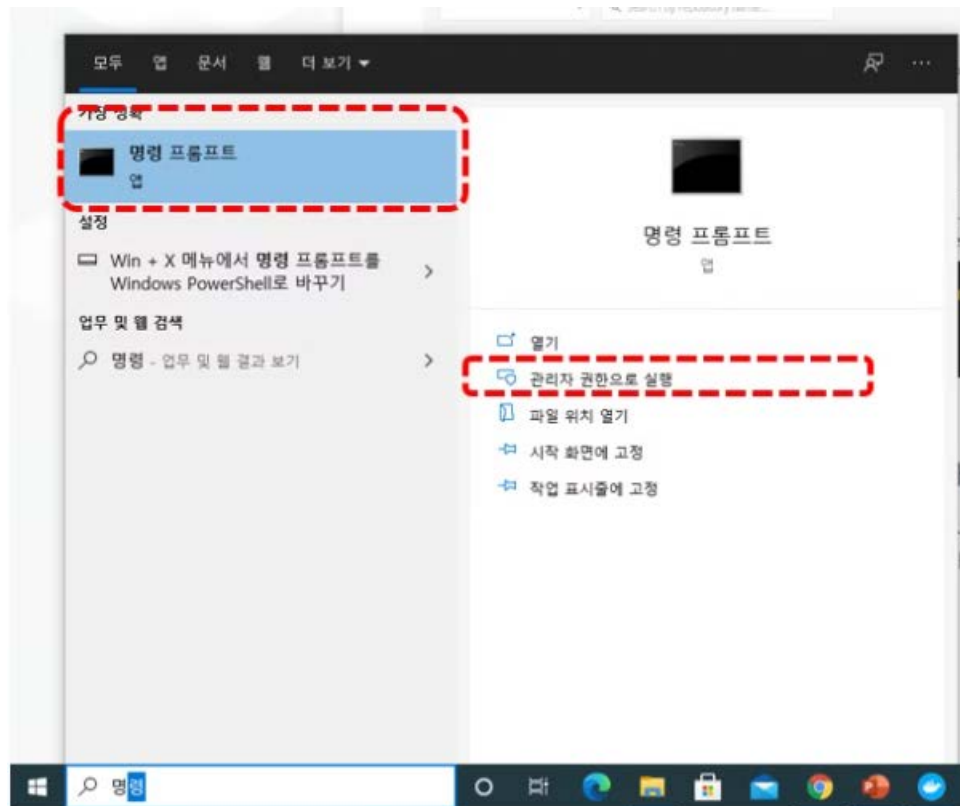


Engine running 확인

Docker 실행

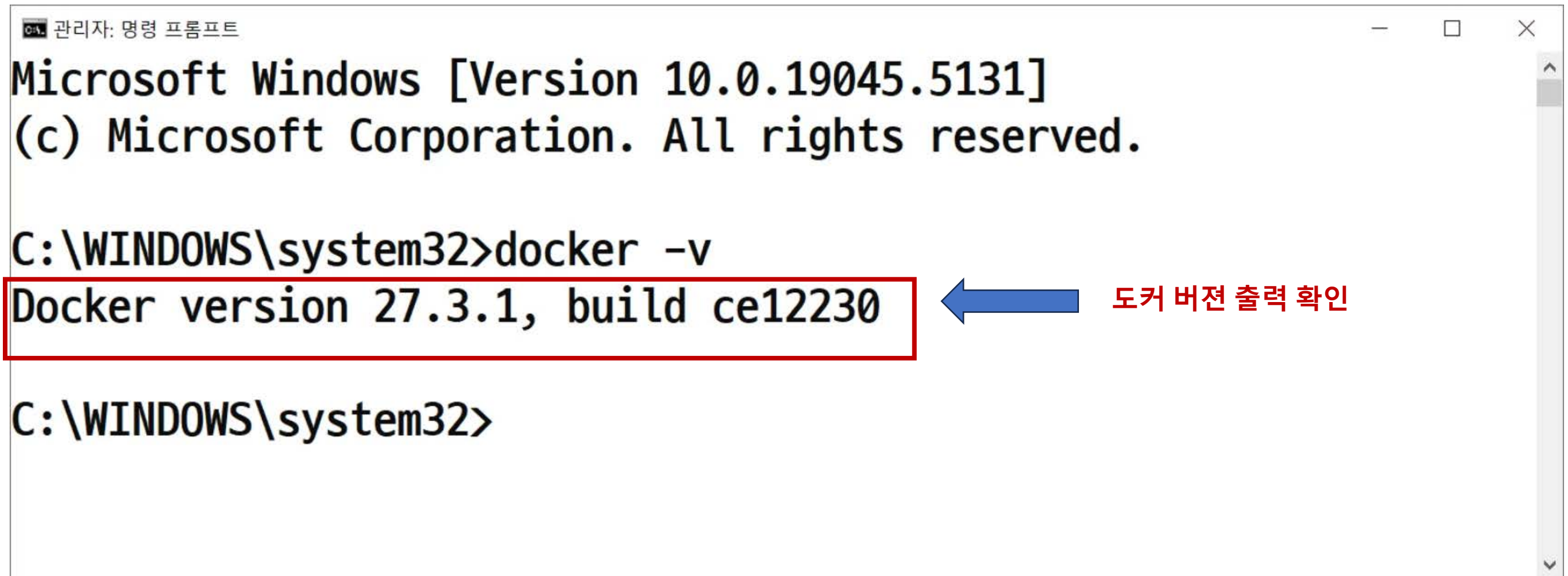
윈도우즈에서 명령 프롬프트 검색

명령 프롬프트 -> 관리자 권한으로 실행 클릭



Docker 실행

docker -v 입력해서 Docker 버전이 출력되는지 확인



```
C:\> 관리자: 명령 프롬프트
Microsoft Windows [Version 10.0.19045.5131]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>docker -v
Docker version 27.3.1, build ce12230

C:\WINDOWS\system32>
```

← 도커 버전 출력 확인

4.Docker 가상화 특징

Docker는 컨테이너로 애플리케이션과 종속성을 격리하여 가상화합니다.

이를 통해 리소스를 공유하며 운영 체제 수준에서 가벼운 배포와 빠른 실행이 가능합니다.

Docker 가상화를 이용한 Ubuntu 실행

Docker의 핵심 기능은 가상화입니다. 가상화란 하나의 물리적 컴퓨터를 여러 가상의 컴퓨터처럼 사용할 수 있게 만드는 기술입니다. Docker는 가상화 기술을 기반으로 작동하지만, 기존의 가상화 방식과는 차별화된 컨테이너 기반 가상화를 제공합니다. 이를 통해 훨씬 더 가볍고 빠르게 환경을 만들고 배포할 수 있습니다.

현재 사용하는 Windows는 본래의 식당(운영체제)입니다. 그런데 가끔 손님이 Windows 식당으로는 만들기 어려운 "Ubuntu 식당에서 만든 요리"를 주문할 수 있습니다. 이 경우, 새로운 식당(새 운영체제)을 실제로 설치하지 않고도, Docker라는 기술을 통해 간단히 가상의 식당 (Container) Ubuntu 을 만들고 "Ubuntu 식당에서 요리"를 수행 할 수 있습니다.

가상화 기술의 진화:

Docker와 기존 가상화의 차이점

기존 가상화 (Virtual Machine)

- 새로운 운영체제 설치 필요: 각 가상 컴퓨터마다 운영체제를 새로 설치해야 합니다.
- 무겁고 자원을 많이 사용: 운영체제 전체를 설치하므로 CPU, 메모리, 디스크 공간이 많이 필요합니다.
- 독립적이지만 비효율적: 각 가상 컴퓨터가 별도의 운영체제를 가지고 있어 비효율적입니다.
- 비유: 각 가상 식당마다 냉장고, 오븐 등 모든 설비를 처음부터 끝까지 따로 설치 해야 하기 때문에 공간과 시간이 많이 필요

Docker (컨테이너 기반 가상화)

- 운영체제를 공유: 물리적 서버에 설치된 운영체제를 그대로 활용하고 필요한 환경만 추가합니다.
- 가볍고 자원을 적게 사용: 필요한 최소한의 도구와 설정만 포함하므로 시스템 부하가 적습니다.
- 효율적이고 빠른 실행: 기존 운영체제 위에서 바로 실행되므로 시작 속도가 매우 빠릅니다.
- 비유: 기존 식당의 냉장고와 오븐을 공유하면서, 필요한 재료와 도구만 추가로 가져와 요리를 만드는 방식입니다. 설비를 새로 설치하지 않으므로 훨씬 가볍고 빠릅니다.

5.Docker 가상환경 ubuntu 실행

Docker를 사용하면 자원 사용이 적고 빠른 속도의 Ubuntu 가상 환경을 구현할 수 있습니다.

Docker 가상환경 ubuntu 실행

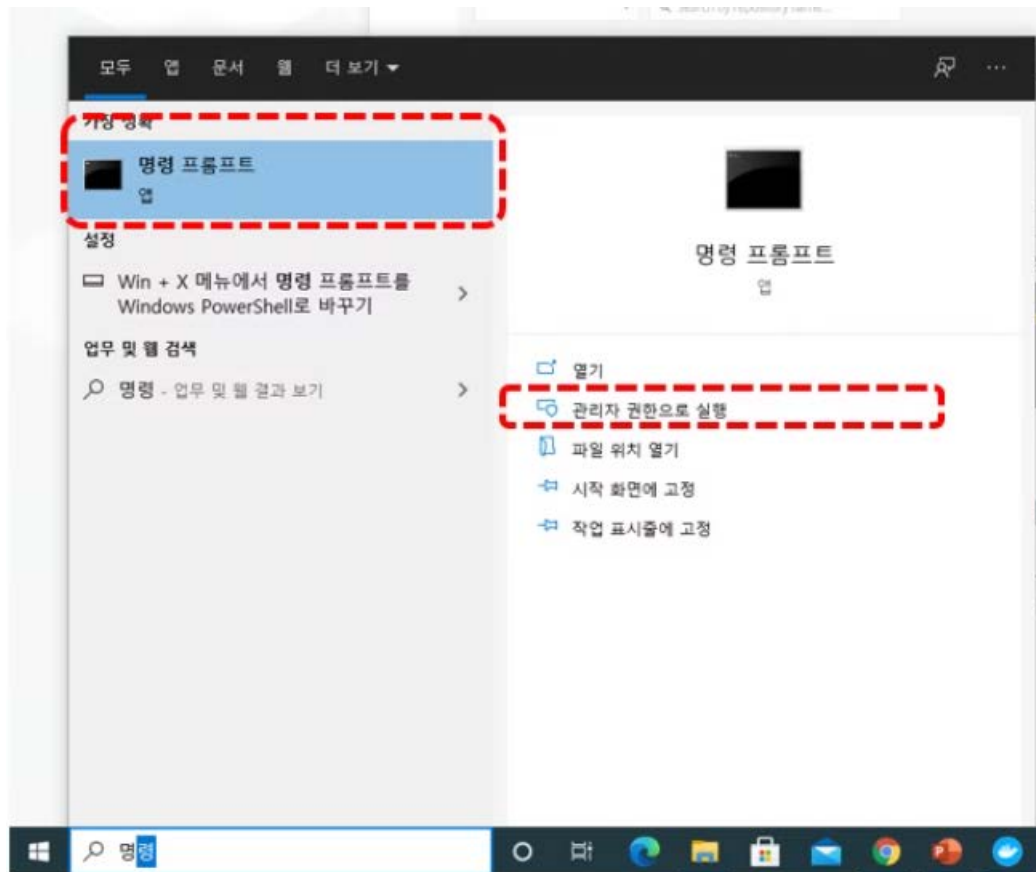
Docker는 컴퓨터 안에서 가상의 식당을 만드는 기술입니다.

이 가상의 식당에는 우리가 필요한 모든 조리 도구와 환경이 준비되어 있어서, 어디서나 집에서와 똑같은 밥을 만들 수 있습니다. 즉 다른 컴퓨터나 서버에서도 똑같은 환경으로 프로그램을 실행 시킬수 있습니다.

Docker 가상환경 ubuntu 실행

윈도우즈에서 명령 프롬프트 검색

명령 프롬프트 -> 관리자 권한으로 실행 클릭



Docker 이용 가상 식당 Ubuntu 에서 요리하기

1

식당 설계도 가져오기

docker pull 명령어를 사용하여 식당 설계도를 다운로드합니다.

2

가져온 설계도 리스트 조회

docker images 명령어로 다운로드된 이미지를 확인합니다.

3

식당 만들고 요리 시작

docker run 명령어로 컨테이너를 시작하여 요리를 시작합니다.

4

식당 상태 확인

docker ps 명령어로 현재 실행 중인 컨테이너를 검사합니다.

5

요리 중단

docker stop 명령어로 요리를 일시 중지합니다.

DOCKER HUB



docker pull

가상 식당의 설계도(이미지) 가져오기

가상 식당을 만들기 위해서 가상 식당의 설계도를 가져 옵니다.

Ubuntu 가상 식당의 설계도 가져오기 (docker pull)

```
docker pull ubuntu
```

설계도 (이미지) 가져오기

Docker가 제공하는 공식 저장소(설계도를 찾는 도서관)에서 원하는 시스템(예: Ubuntu)의 설계도를 가져오는 명령어

다운로드 완료 후

이는 식당을 짓기위해서 설계도를 가져오는 것과 같습니다. 다운로드가 완료되면, "Ubuntu 이미지"라는 설계도가 준비됩니다.

```
관리자: 명령 프롬프트

C:\WINDOWS\system32>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
ff65ddf9395b: Download complete
Digest: sha256:99c35190e22d294cdace2783ac55effc69d32896daaa265f0bbedbcde4fbe3e5
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

C:\WINDOWS\system32>
```



docker images

가지고 있는 설계도 리스트 확인

다운로드 한 설계도 목록 확인

docker images

설계도 확인

다운로드한 Docker 이미지 목록을 확인할 수 있습니다. 이를 통해 현재 시스템에 어떤 설계도가 준비되어 있는지 확인할 수 있습니다.

이미지 관리

이미지 목록을 확인하면 사용하지 않는 이미지를 정리하거나, 새로운 이미지를 추가할 수 있습니다.

선택 관리자: 명령 프롬프트

C:\WINDOWS\system32>docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	99c35190e22d	5 weeks ago	117MB

C:\WINDOWS\system32>



docker run

다운받은 설계도로 식당을 짓고 요리 시작

Docker 식당 짓고 요리 시작

`docker run -it ubuntu`

다운받은 설계도로 식당을 짓고 요리 시작

다운 받은 설계도 (이미지) 로 Ubuntu 식당을 짓고 요리 시작

`-it` 옵션: 사용해 대화형 모드로 진입, 직접 식당에서 작업할 수 있습니다.

터미널 접근

`root@컨테이너ID` 형식이 보이면 Ubuntu 식당 안에 들어간 상태로 직접 요리를 시작 할 수 있습니다.

```
root@721cf06ed467: /  
C:\WINDOWS\system32> docker run -it ubuntu  
root@721cf06ed467: /#
```




가상 식당 ubuntu 안에 들어감

pwd

사용자가 들어가 있는 가상 식당 Ubuntu 가상 환경에서 우분투 pwd 명령어로 현재 위치를 확인합니다.

```
root@8c263e93415c: /  
  
C:\WINDOWS\system32>docker run -it ubuntu  
root@8c263e93415c: /# pwd  
/  
root@8c263e93415c: /#
```



현재 경로 /

ls

사용자가 들어가 있는 가상 식당 Ubuntu 가상 환경에서 우분투 ls 명령어로 현재 위치의 파일 리스트를 확인합니다.

```
root@8c263e93415c: /  
C:\WINDOWS\system32>docker run -it ubuntu  
root@8c263e93415c:/# pwd  
/  
root@8c263e93415c:/# ls  
bin    dev    home  lib64  mnt    proc   run    srv    tmp    var  
boot   etc    lib   media  opt    root   sbin   sys    usr  
root@8c263e93415c:/#
```

↑ 현재 경로 파일 리스트



본 식당으로 돌아가기

가상의 식당 Ubuntu 에서 빠져 나가서 본식당 Windows로 돌아가기 위해 다음과 같이 합니다.

- Ctrl 키와 P 키를 동시에 누릅니다.
- 이어서 Ctrl 키와 Q 키를 동시에 누릅니다.

```
관리자: 명령 프롬프트

C:\WINDOWS\system32>docker run -it ubuntu
root@8c263e93415c:/# pwd
/
root@8c263e93415c:/# ls
bin    dev    home  lib64  mnt    proc  run    srv    tmp    var
boot  etc    lib   media  opt    root  sbin   sys    usr
root@8c263e93415c:/#
C:\WINDOWS\system32>
```

← 본식당 윈도우



docker ps

실행 중인 식당 조회

실행 중인 식당 조회

docker ps

실행 중인 가상 식당 (컨테이너) 리스트와 상태 확인

현재 실행 중인 Ubuntu 식당(컨테이너)의 상태를 확인할 수 있습니다.

각 식당은 고유 ID를 가지고 있으며, 필요하면 이 ID를 통해 제어할 수 있습니다.

```
C:\WINDOWS\system32>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS          NAMES
8c263e93415c   ubuntu   "/bin/bash"             20 minutes ago  Up 20 minutes  sharp_hermann

C:\WINDOWS\system32>
```



docker attach

가상 식당으로 돌아 가기

본 식당에서 가상 식당으로 돌아가기

`docker attach [가상 식당 ID]`

실행 중인 가상 식당 (컨테이너) 으로 돌아가기

본 식당 Windows에서 현재 실행 중인 Ubuntu 식당(컨테이너)으로 돌아 갑니다.

```
C:\WINDOWS\system32>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
8c263e93415c   ubuntu    "/bin/bash"             20 minutes ago Up 20 minutes                sharp_hermann

C:\WINDOWS\system32>docker attach 8c263e93415c
root@8c263e93415c:/# ls
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var
boot  etc  lib  media  opt  root /sbin  sys  usr
root@8c263e93415c:/#
```

컨테이너 ID 입력



본 식당으로 돌아가기

가상의 식당 Ubuntu 에서 빠져 나가서 본식당 Windows로 돌아가기 위해 다음과 같이 합니다.

- Ctrl 키와 P 키를 동시에 누릅니다.
- 이어서 Ctrl 키와 Q 키를 동시에 누릅니다.

```
관리자: 명령 프롬프트

C:\WINDOWS\system32>docker run -it ubuntu
root@8c263e93415c:/# pwd
/
root@8c263e93415c:/# ls
bin    dev    home  lib64  mnt    proc   run    srv    tmp    var
boot  etc    lib   media  opt    root   sbin   sys    usr
root@8c263e93415c:/#
C:\WINDOWS\system32>
```

← 본식당 윈도우



docker stop

현재 요리를 멈추기

실행 중인 식당 종료

`docker stop [컨테이너 ID]`

실행 중인 가상 식당 (컨테이너) 종료

Ubuntu 식당의 실행을 멈춥니다.

```
관리자: 명령 프롬프트
C:\WINDOWS\system32>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
8c263e93415c  ubuntu   "/bin/bash"  36 minutes ago  Up 36 minutes           sharp_hermann

C:\WINDOWS\system32>docker stop 8c263e93415c
8c263e93415c

C:\WINDOWS\system32>
```

← 컨테이너 ID 입력



docker rm

사용이 끝난 식당 삭제

사용이 끝난 식당 삭제

docker rm [컨테이너 ID]

사용이 끝난 가상 식당 (컨테이너) 삭제

더 이상 사용하지 않는 컨테이너(식당)를 삭제하여 공간을 확보합니다.

```
C:\WINDOWS\system32>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8c263e93415c	ubuntu	"/bin/bash"	42 minutes ago	Exited (137) 5 minutes ago		sharp_hermann

```
C:\WINDOWS\system32>docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8c263e93415c	ubuntu	"/bin/bash"	42 minutes ago	Exited (137) 5 minutes ago		sharp_hermann

```
C:\WINDOWS\system32>docker rm 8c263e93415c
8c263e93415c
```

```
C:\WINDOWS\system32>docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
C:\WINDOWS\system32>
```

← 사용 중인 식당만 조회

← -a 모든 식당 조회

← 컨테이너 ID 입력

← 식당이 삭제 되서 조회 안됨