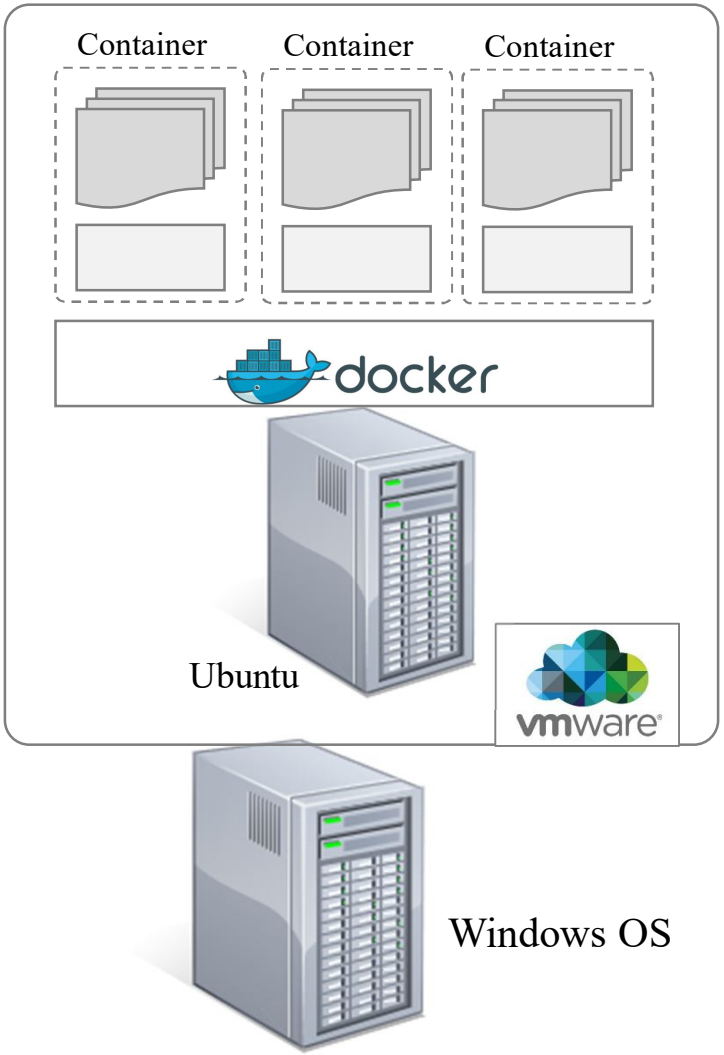
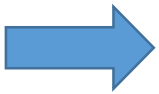
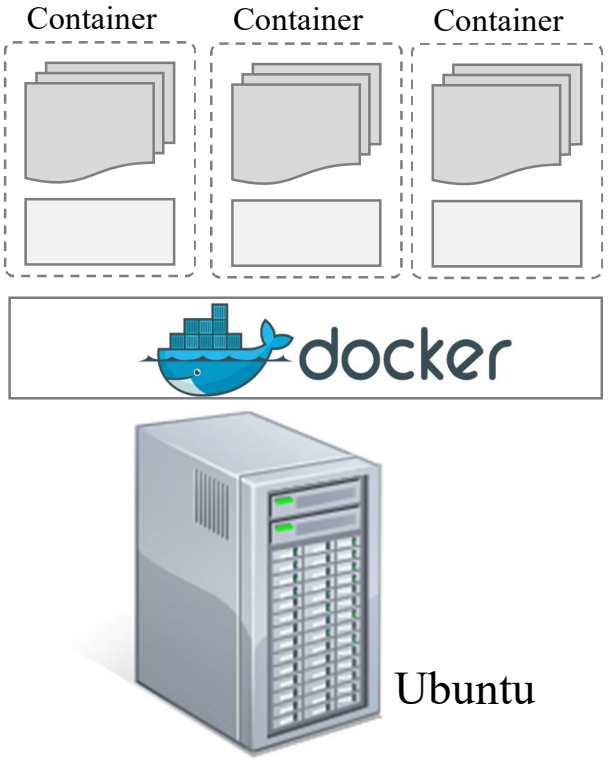
Docker 실습환경 구성

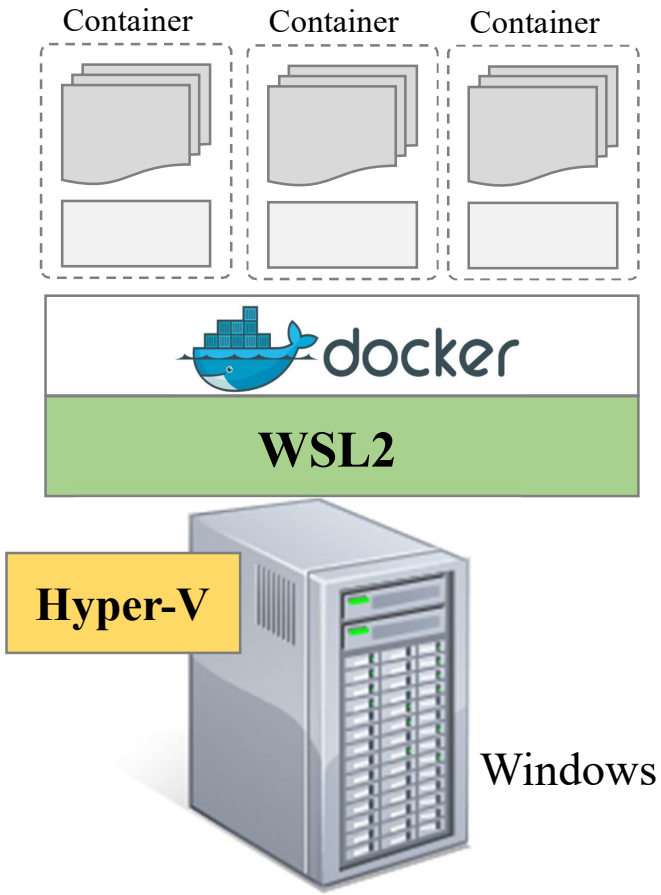
Linux 환경에서 Docker 설치

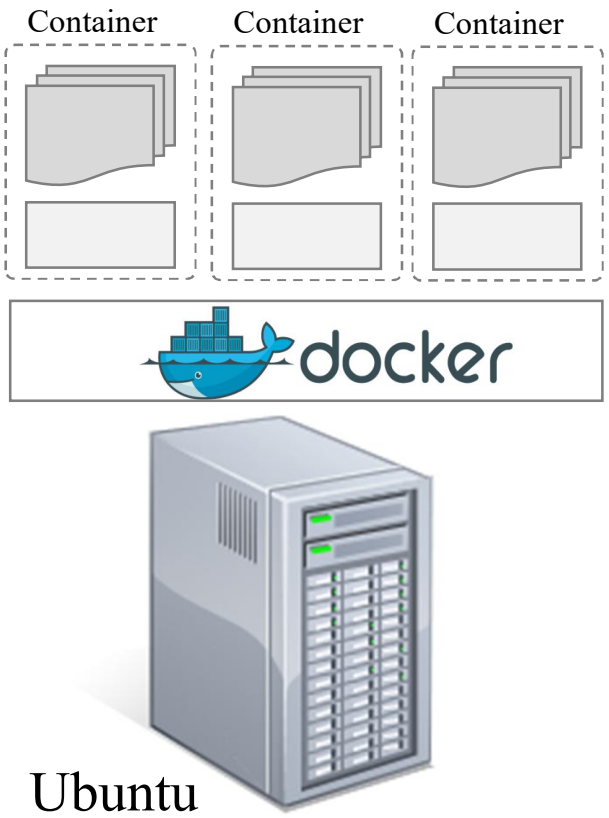


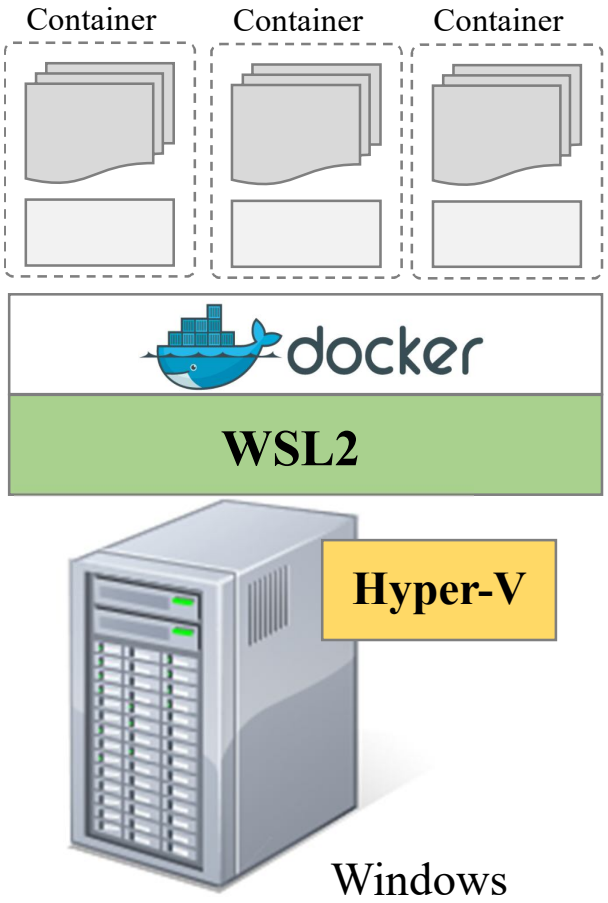




Windows 환경에서 Docker 설치




Container
Container
Container
docker
Ubuntu

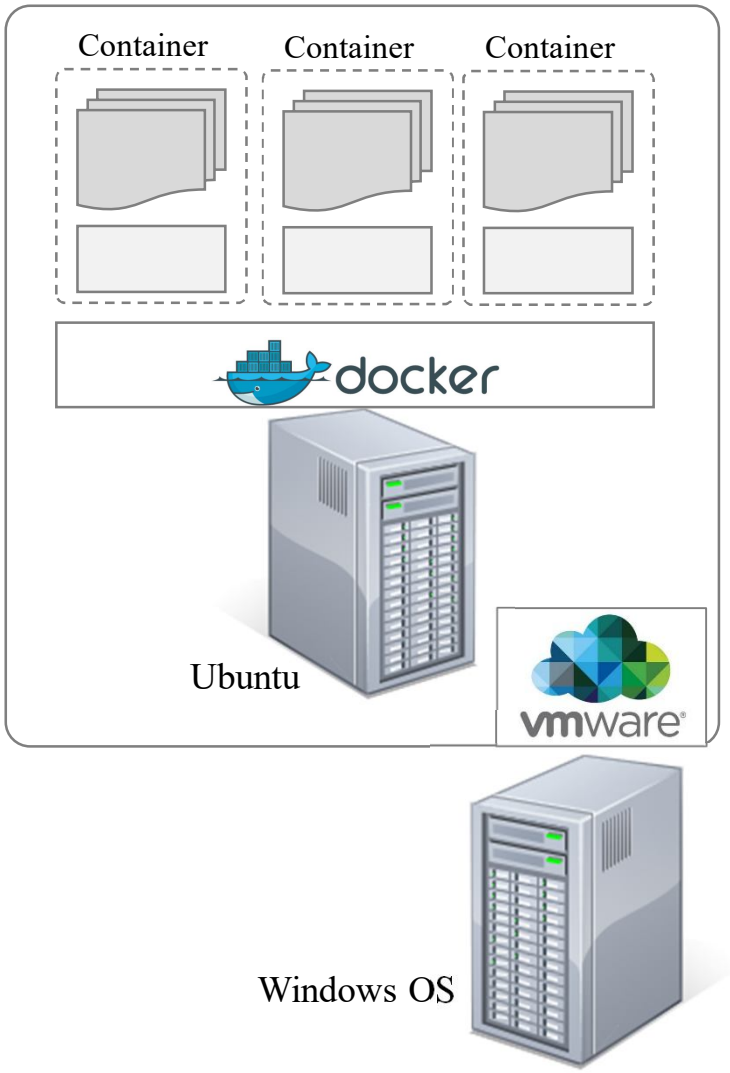

Container
Container
Container
docker
WSL2
Hyper-V
Windows

Linux 환경에서 Docker 설치

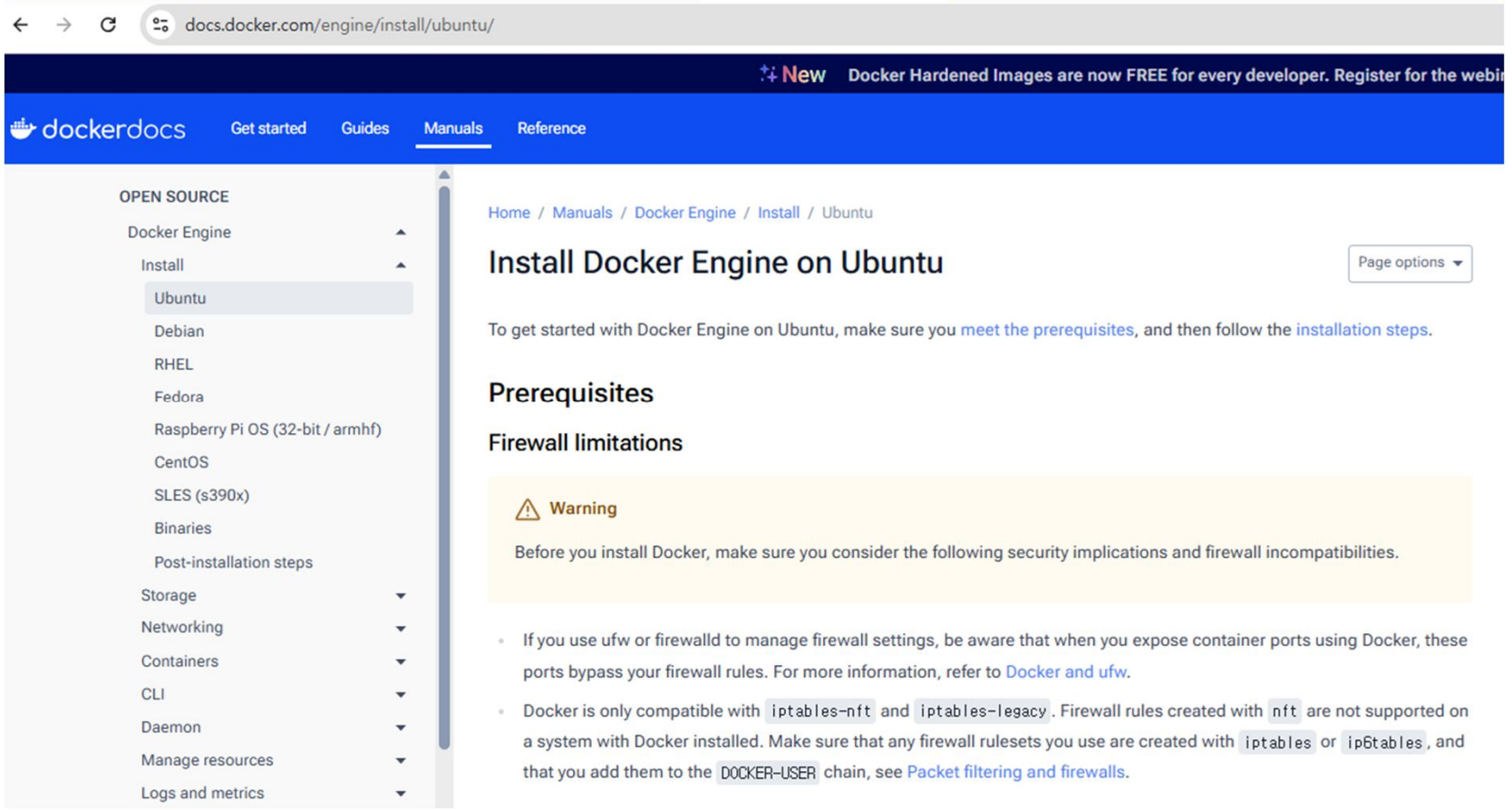
1) VMWare Install

2) Ubuntu Install

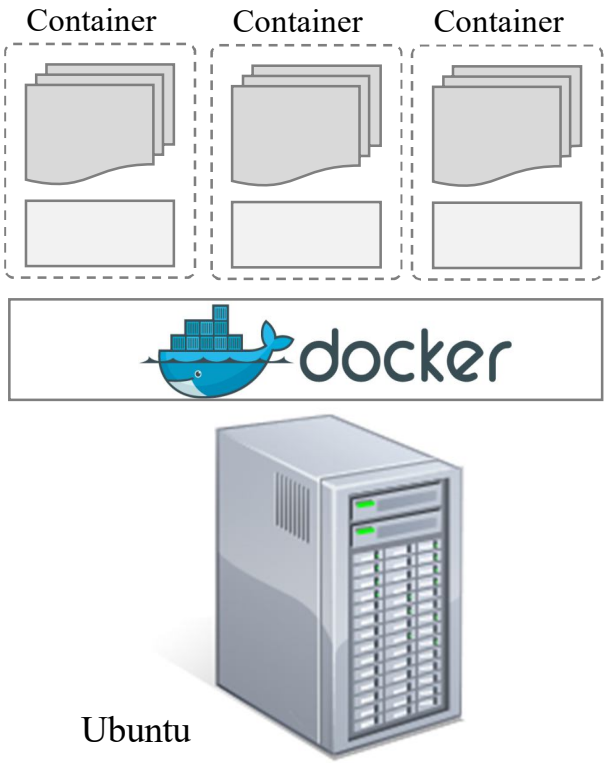
3) Docker Install



https://docs.docker.com/desktop/install/ubuntu/



리눅스 기반의 Docker 설치 방법

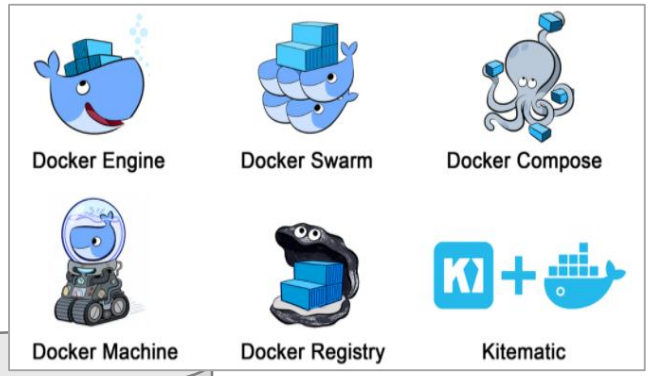


- Repository 를 이용한 설치
- 파일 다운로드 후 설치

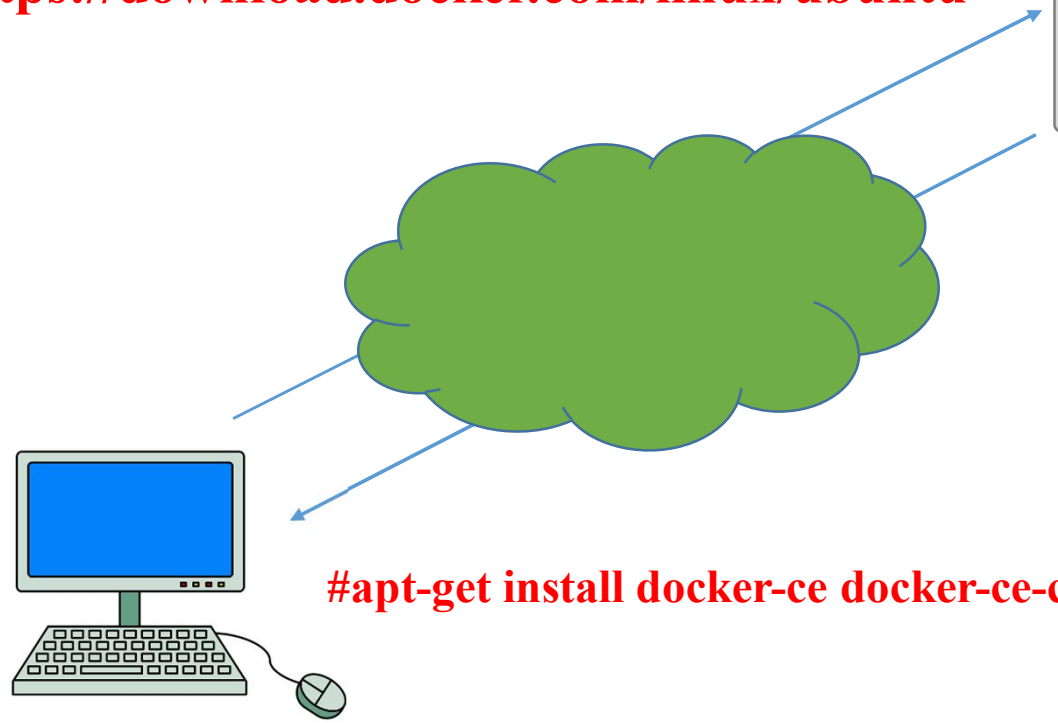
  - 네트워크가 연결이 안 될 경우

- Script를 이용한 설치

Repository를 이용한 Docker Engine 설치



https://download.docker.com/linux/ubuntu



Docker Package Repository
(도커 패키지 저장소)

#apt-get install docker-ce docker-ce-cli containerd.io

1단계. Docker 저장소 지정

❶ 유틸리티 파일 설치

apt-get update

apt-get install -y ca-certificates curl

❷ Docker 공식 GPG key 추가 (도커 인증서 저장)

install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o etc/apt/keyrings/docker.asc

chmod a+r /etc/apt/keyrings/docker.asc

❸ Docker Repository URL 등록

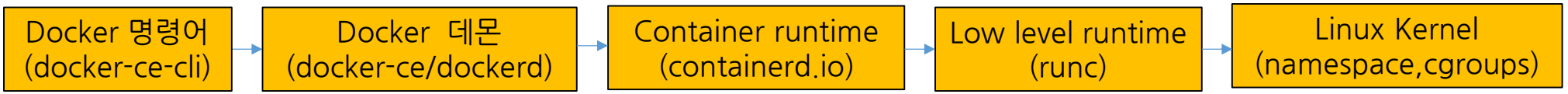
echo “deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu

$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null

apt-get update

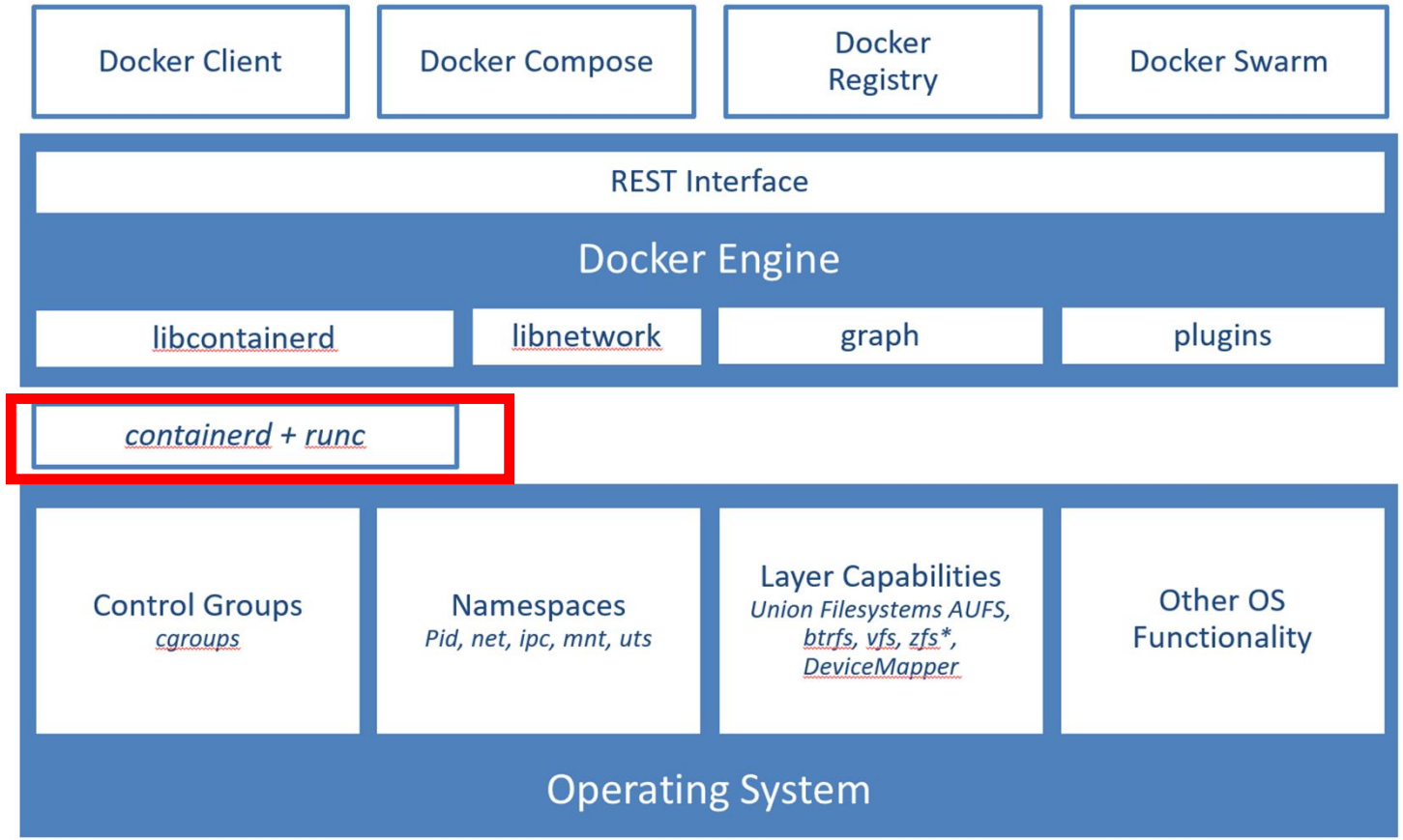
2단계. Docker 패키지 설치

apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

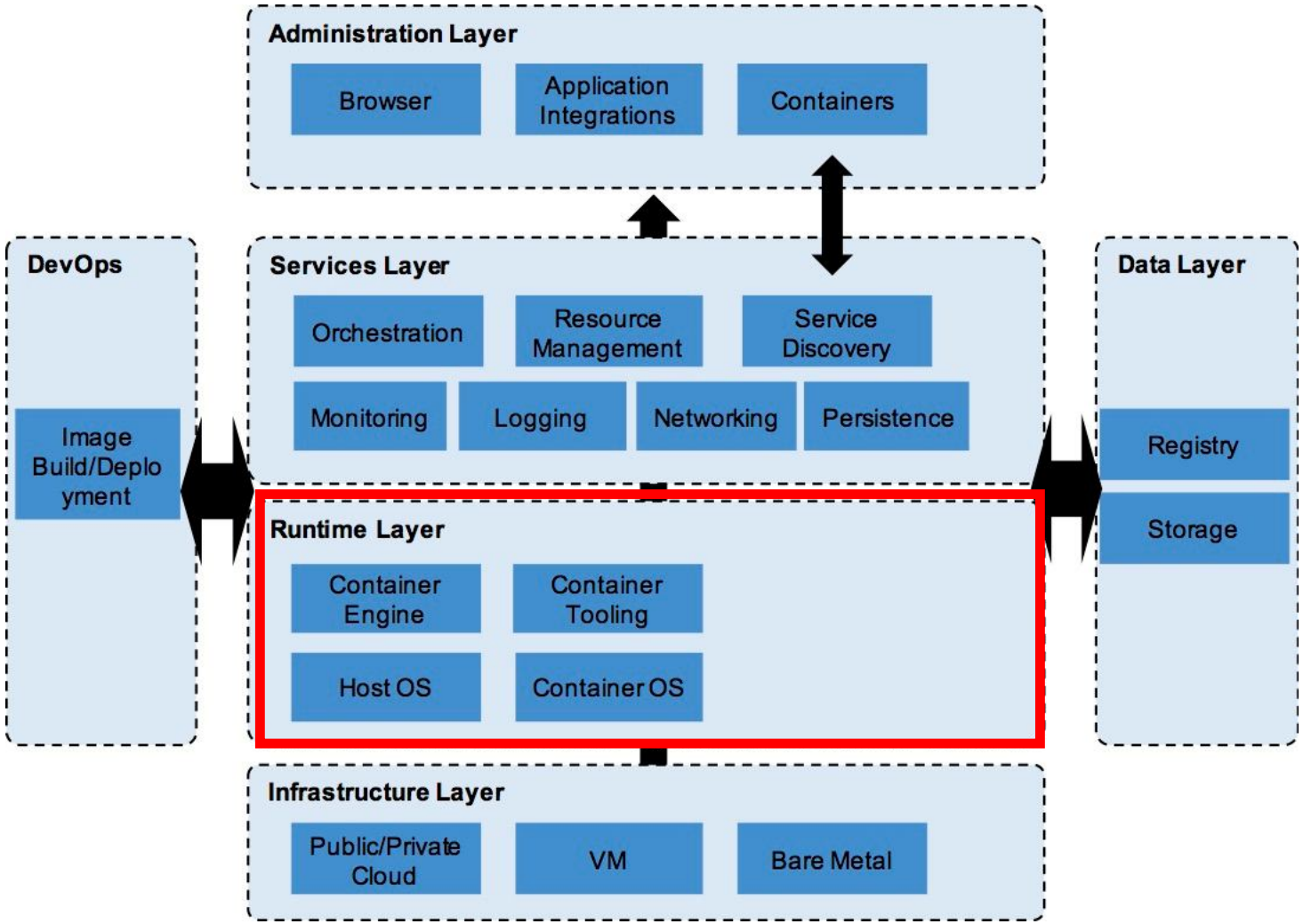


docker-ce-cli	Docker 명령어 도구, docker ps, docker run, docker images 등 사용자와 Docker 데몬(dockerd) 간 인터페이스
docker-ce	Docker Community Edition, Docker 엔진 핵심, dockerd 데몬 제공 컨테이너 생성 / 실행 / 중지 / 삭제, 이미지 관리, 네트워크, 볼륨 제어 systemd 서비스로 동작
containerd.io	컨테이너 런타임 (하부 엔진, 실제 컨테이너 실행 담당) => Docker deamon(프로세스)을 실제로 실행시켜 주는 역할 이미지 pull / push / 저장, 컨테이너 lifecycle 관리, 내부적으로 runc 사용

Architecture In Linux



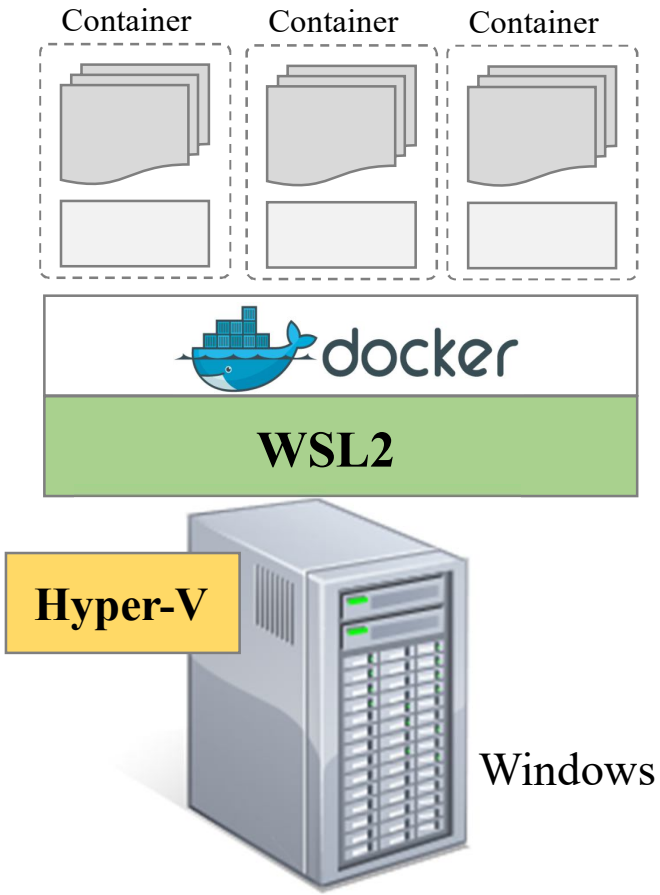
Container Platform Architecture Reference model



3단계. 설치 확인

docker version

Windows 환경에서 Docker 설치



1) Hyper-V 가상화 기능 활성화

2) Hub.docker.com 계정 등록

3) Docker Desktop 설치

- WSL2(Windows Subsystem for Linux v.2)를 통해 리눅스 커널 설치

- Docker 설치

4) Docker 동작 상태 확인

Docker DeskTop

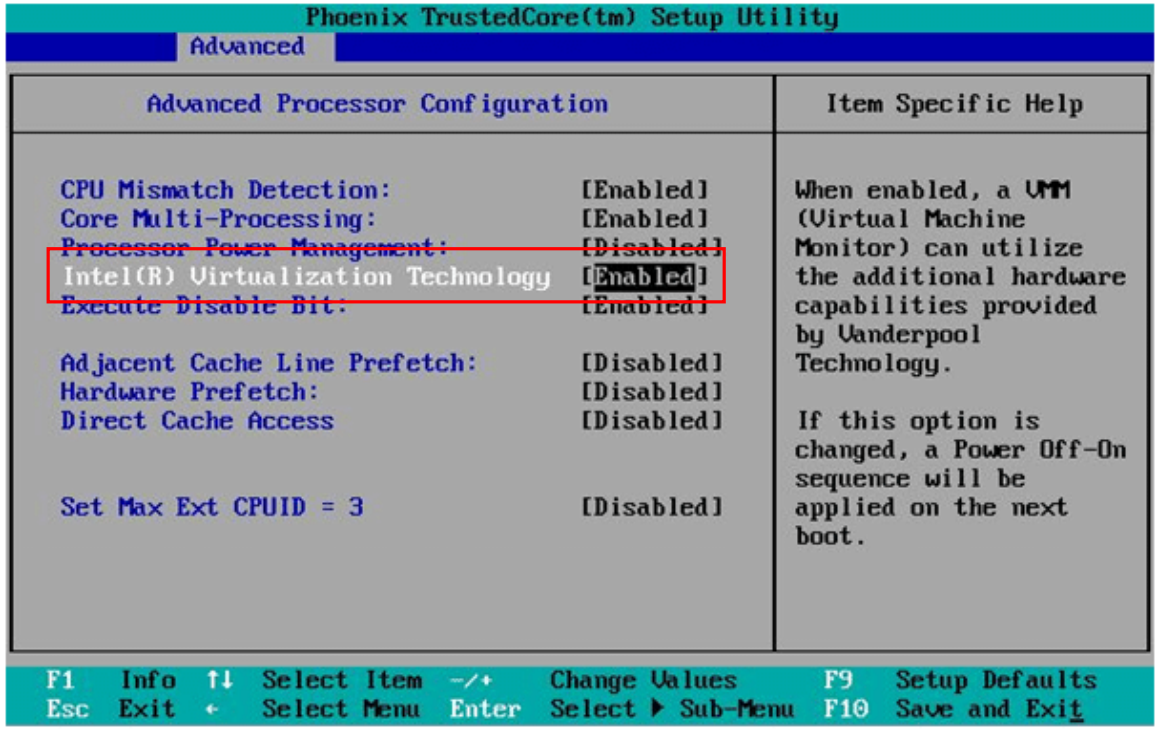
- 컨테이너화된 애플리케이션 및 마이크로서비스를 구축하고 공유할 수 있는 Mac, Linux, Windows 환경용 원클릭 설치 애플리케이션
- 설치된 머신에서 컨테이너, 애플리케이션, 이미지를 관리할 수 있는 간단한 GUI를 제공
- 관련 패키지를 포함

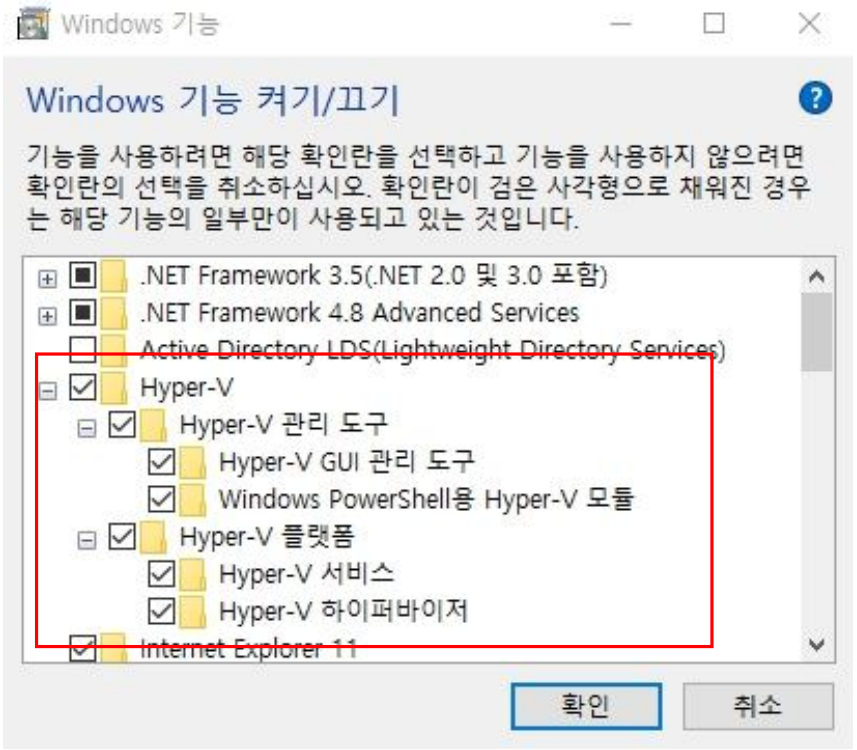
  - Docker Engine, Docker CLI client, Docker Compose, Docker Content Trust, **Kubernetes**, and Credential Helper

WSL2(Windows Subsystem for Linux 2)

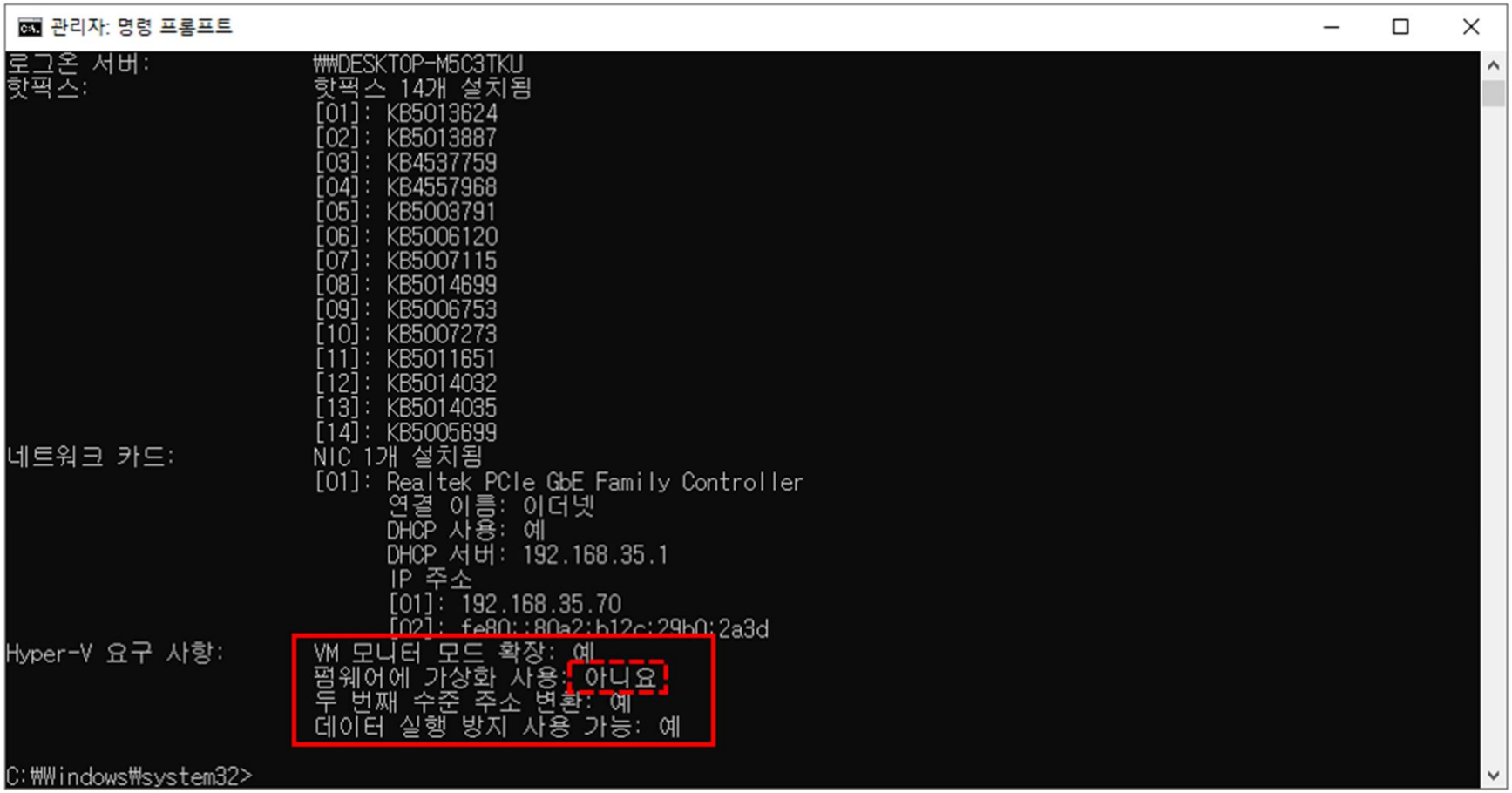
- 윈도우에서 리눅스를 사용할 수 있게 해주는 기능
- 윈도우의 가상화 기능을 활용해서 윈도우 위에서 리눅스를 사용할 수 있게해 줌

1단계. Hyper-V 활성화

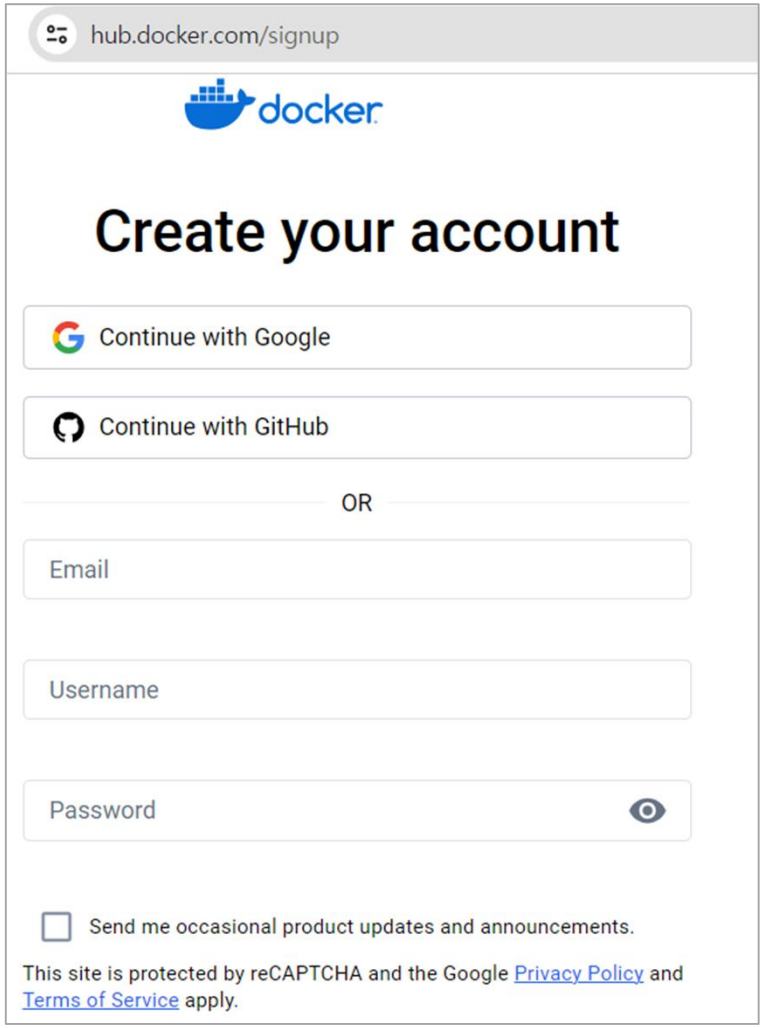




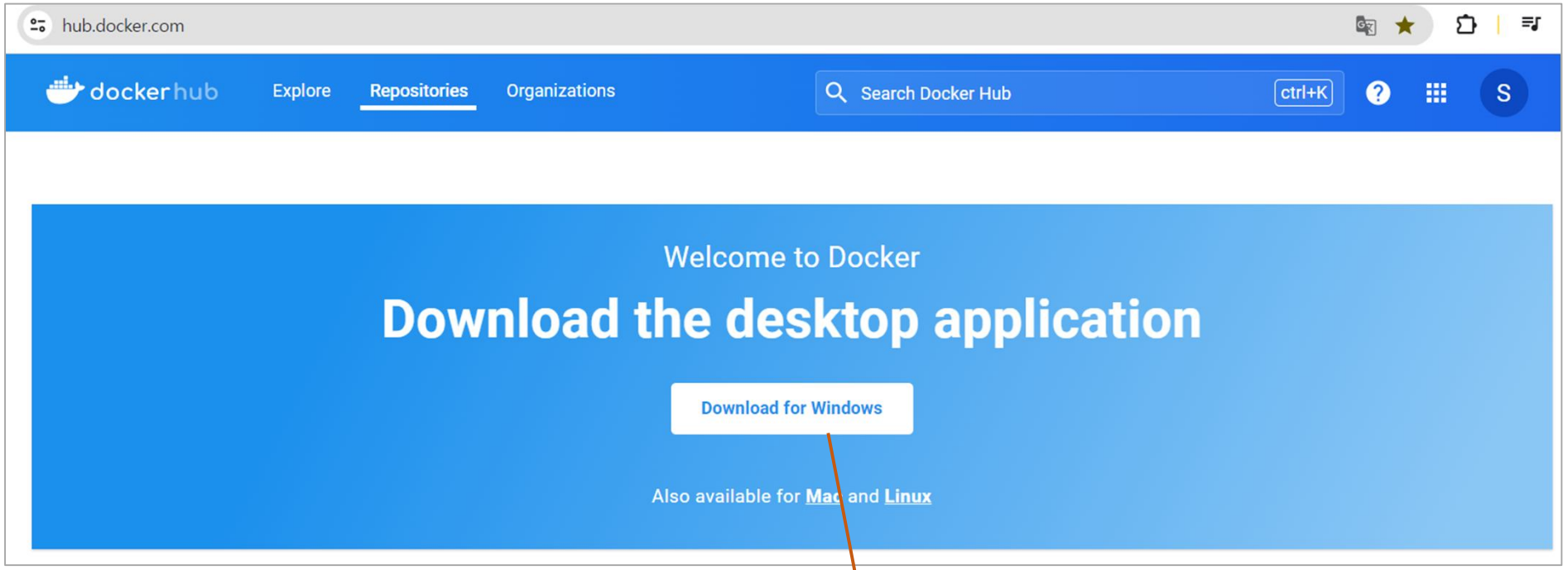
systeminfo

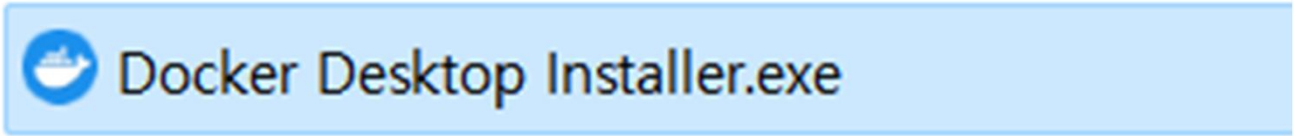

관리자: 명령 프롬프트
로그온 서버: \\DESKTOP-M5C3TKU
핫픽스: 핫픽스 14개 설치됨
[01]: KB5013624
[02]: KB5013887
[03]: KB4537759
[04]: KB4557968
[05]: KB5003791
[06]: KB5006120
[07]: KB5007115
[08]: KB5014699
[09]: KB5006753
[10]: KB5007273
[11]: KB5011651
[12]: KB5014032
[13]: KB5014035
[14]: KB5005699
네트워크 카드: NIC 1개 설치됨
[01]: Realtek PCIe GbE Family Controller
연결 이름: 이더넷
DHCP 사용: 예
DHCP 서버: 192.168.35.1
IP 주소
[01]: 192.168.35.70
[02]: fe80::80a2:b12c:29b0:2a3d
Hyper-V 요구 사항: VM 모니터 모드 확장: 예
펌웨어에 가상화 사용: 아니요
두 번째 수준 주소 변환: 예
데이터 실행 방지 사용 가능: 예
C:\Windows\system32>

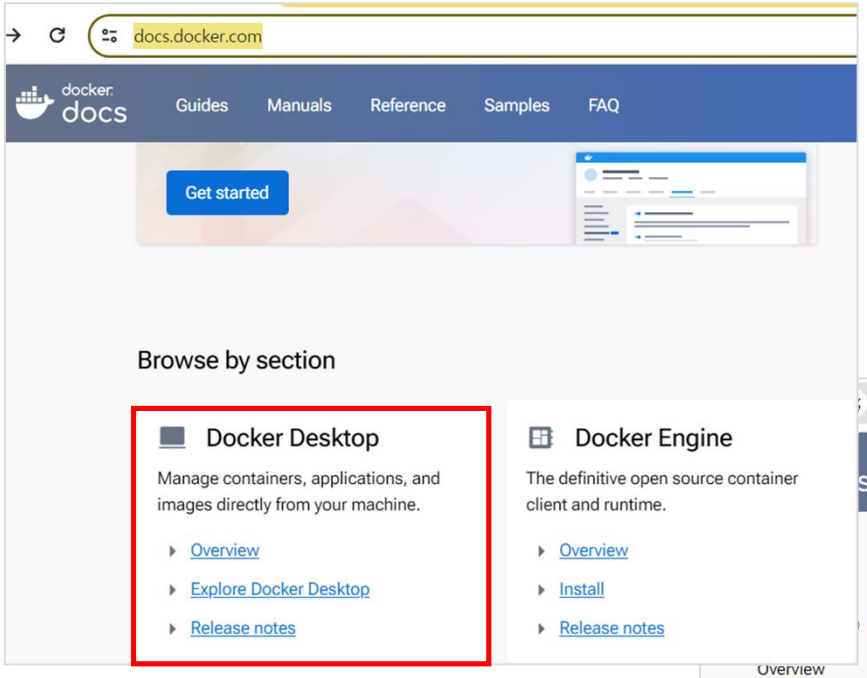
2단계. Hub.docker.com 계정 등록



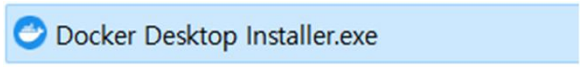
3단계. Docker Desktop 다운로드

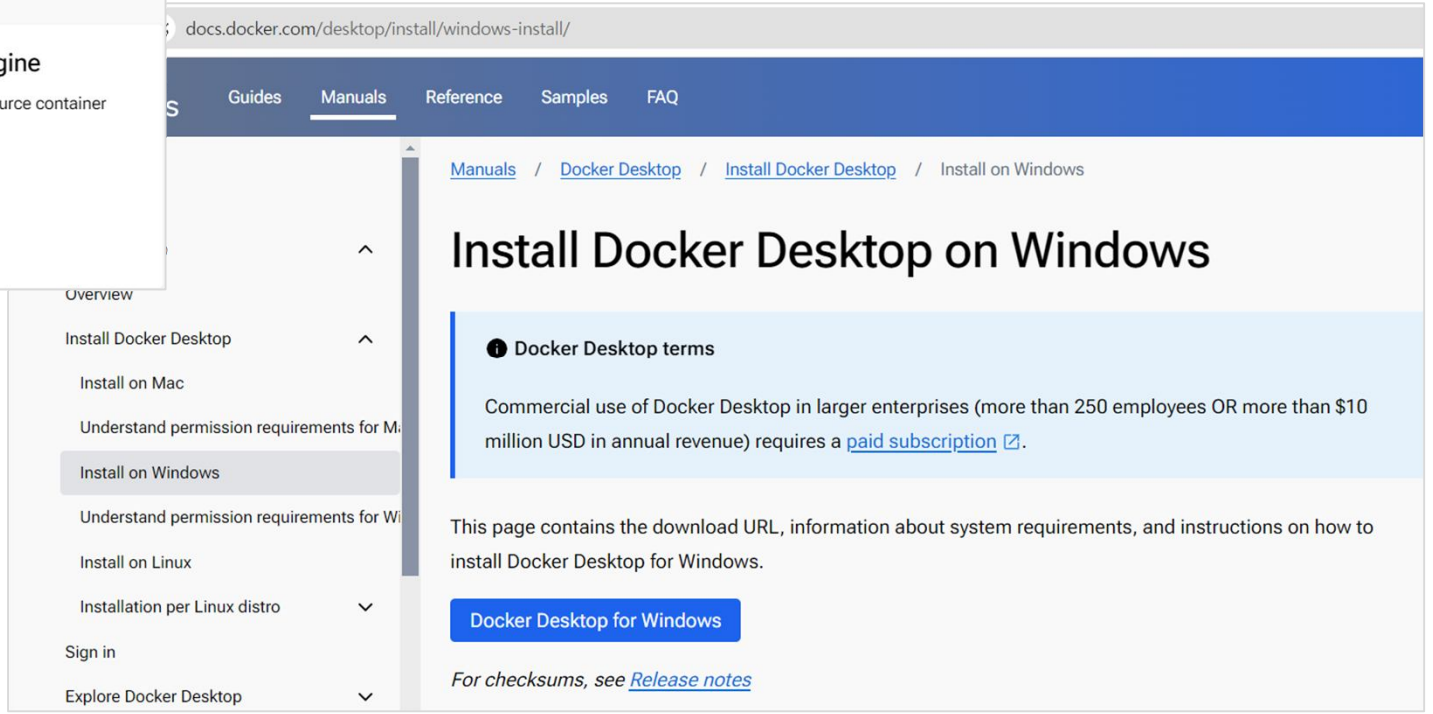




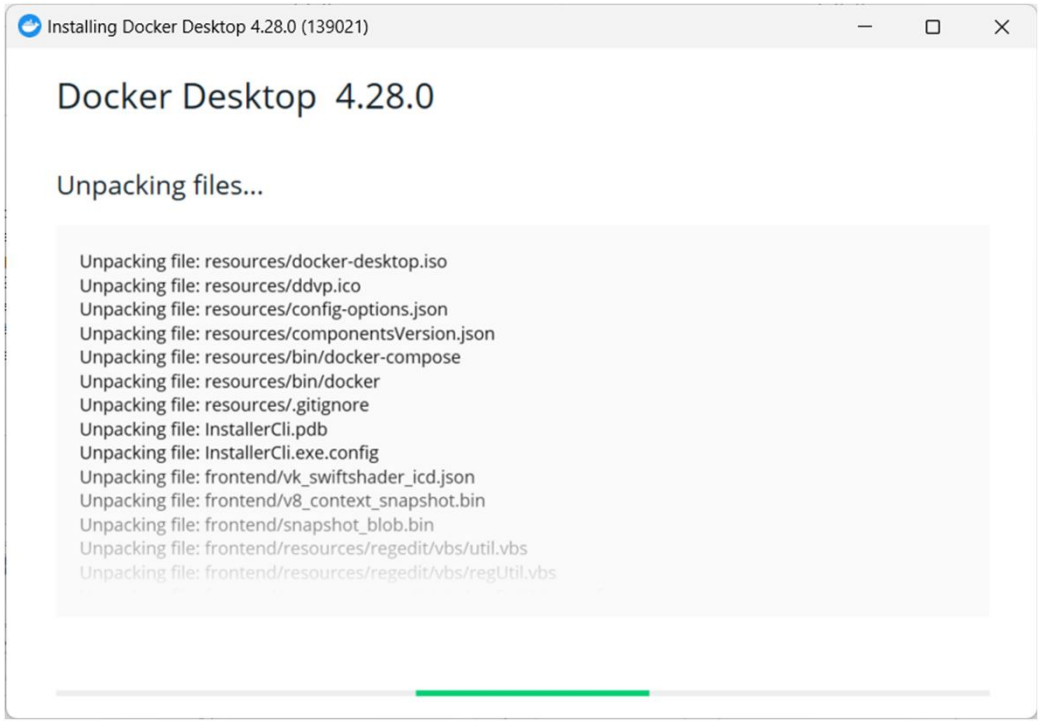


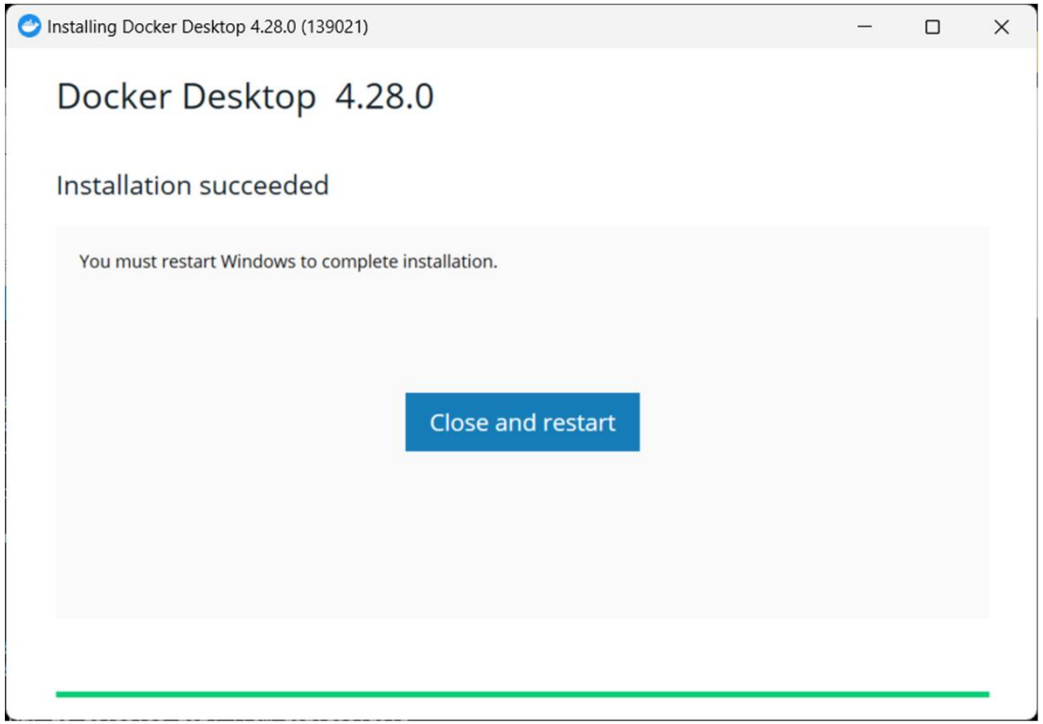
https://docs.docker.com/desktop/install/windows-install/



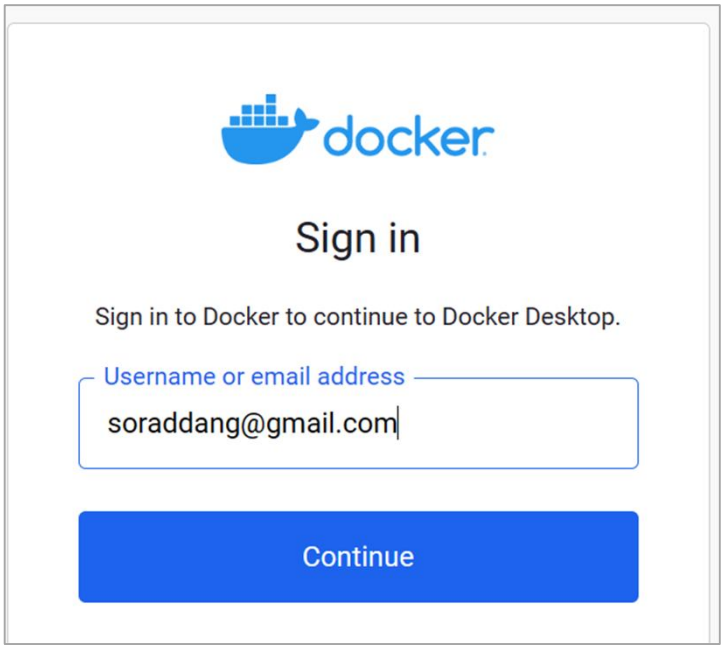


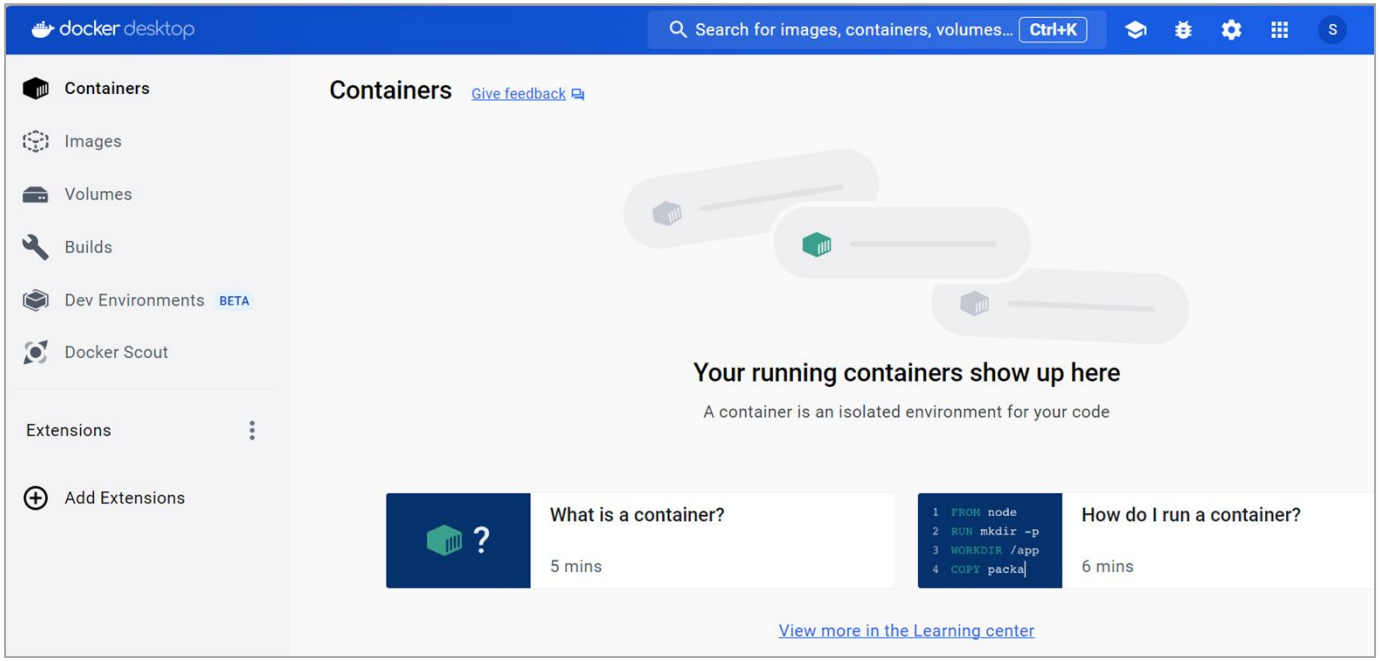
4단계. Docker Desktop 설치





5단계. Docker Hub 로그인 후 Docker Engine 활성화





6단계. PowerShell을 이용한 Docker 확인

```
PS C:\> docker version
Client:
 Cloud integration: v1.0.35+desktop.11
 Version:           25.0.3
 API version:       1.44
 Go version:        go1.21.6
 Git commit:        4debf41
 Built:             Tue Feb  6 21:13:02 2024
 OS/Arch:           windows/amd64
 Context:           default

Server: Docker Desktop 4.28.0 (139021)
 Engine:
  Version:          25.0.3
  API version:      1.44 (minimum version 1.24)
  Go version:       go1.21.6
  Git commit:       f417435
  Built:            Tue Feb  6 21:14:25 2024
  OS/Arch:          linux/amd64
```