

Docker Seminar

Contents

1. Docker의 개념
2. Docker 사용하기
3. Docker-compose

Docker

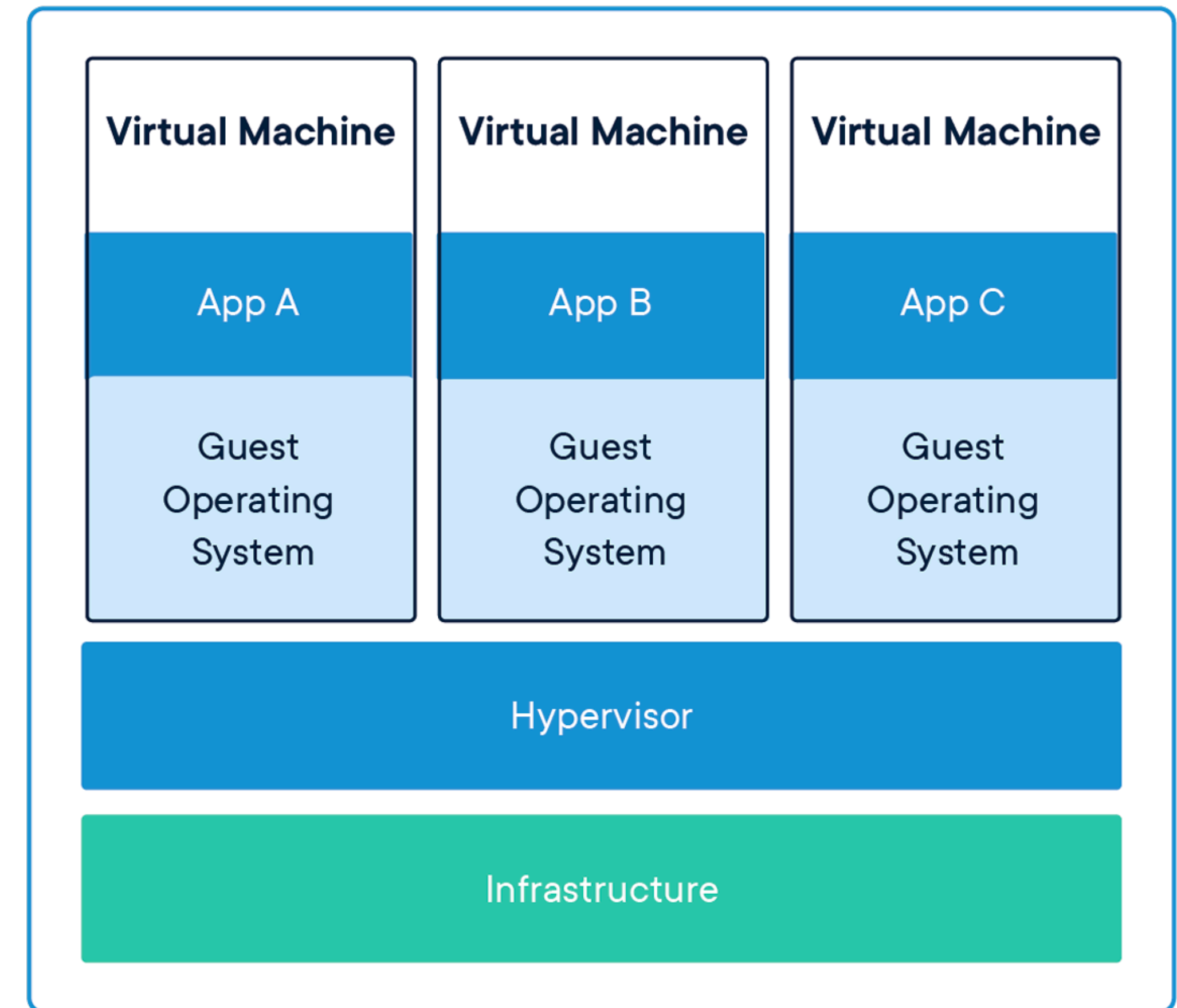
Docker는 Open source container project!

1. Linux 기반의 container runtime tool이다.
2. 쉽게 얘기해서 가상머신 같은 것으로 이해하고 넘어가자

Docker

가상머신

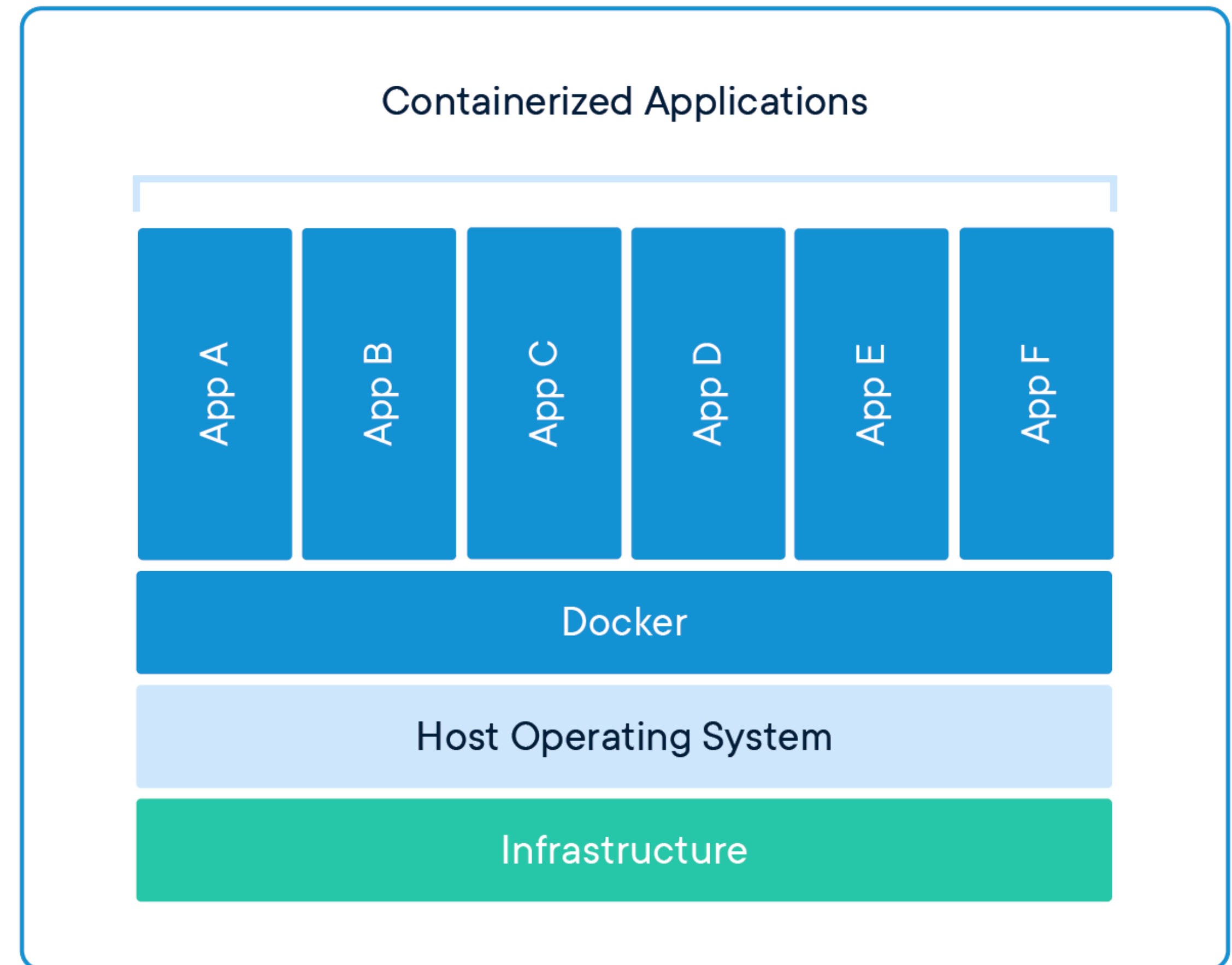
1. Full virtualization, para virtualization
2. Host PC위에 Hypervisor(VMWare, KVM, ...)를 두고, Virtual machine을 올린다.
3. 일종의 하드웨어 가상화
4. 가상머신끼리는 독립적
5. 독립적인 Guest OS 부분을 가지기에 무거움
6. 느린 성능



Docker

컨테이너

1. kernel은 host PC의 것을 사용
2. Host PC와 다른 부분을 container에 올림
3. 실제 command는 host에서 실행됨
4. container끼리는 독립적
5. 독립적인 OS를 사용하지 않음
6. Git과 같은 버전관리가 됨
7. 가벼움
8. 도커는 컨테이너를 사용



Docker

기존의 배포 환경

1. 배포 환경에 맞게 OS 설치하기
2. Web server 혹은 기타 라이브러리 설치하기
3. 데이터베이스 설정 혹은 remote 데이터베이스 설정
4. 소스 코드 복사 및 바이너리 파일 복사
5. 배포
6. 실패
7. 수정
8. 배포
9. 실패
10. ...

Docker

도커를 이용한 배포방법

1. 개발 완료 후 배포 환경에 맞춰 도커 이미지 build
2. 테스트
3. 배포

Docker

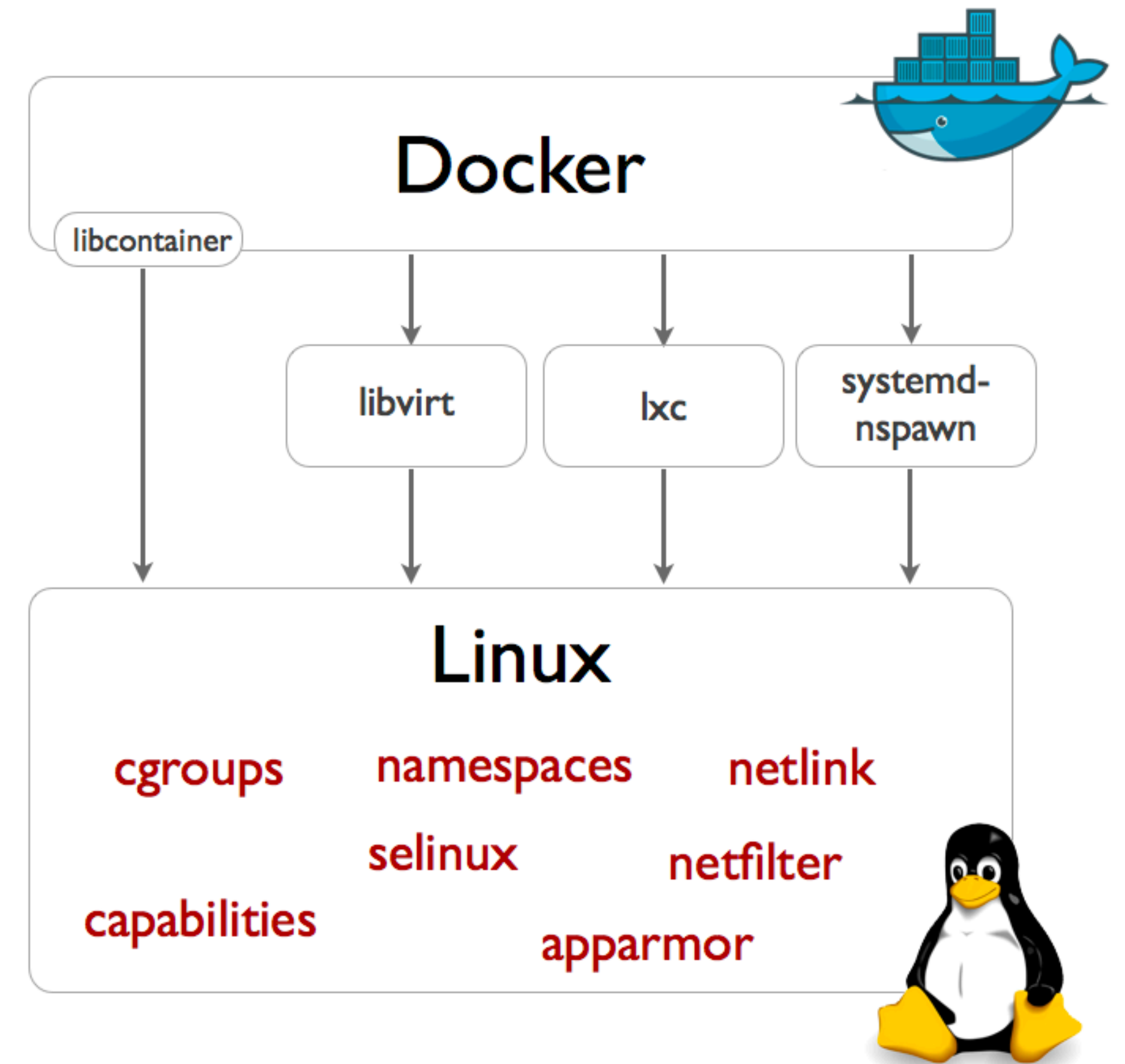
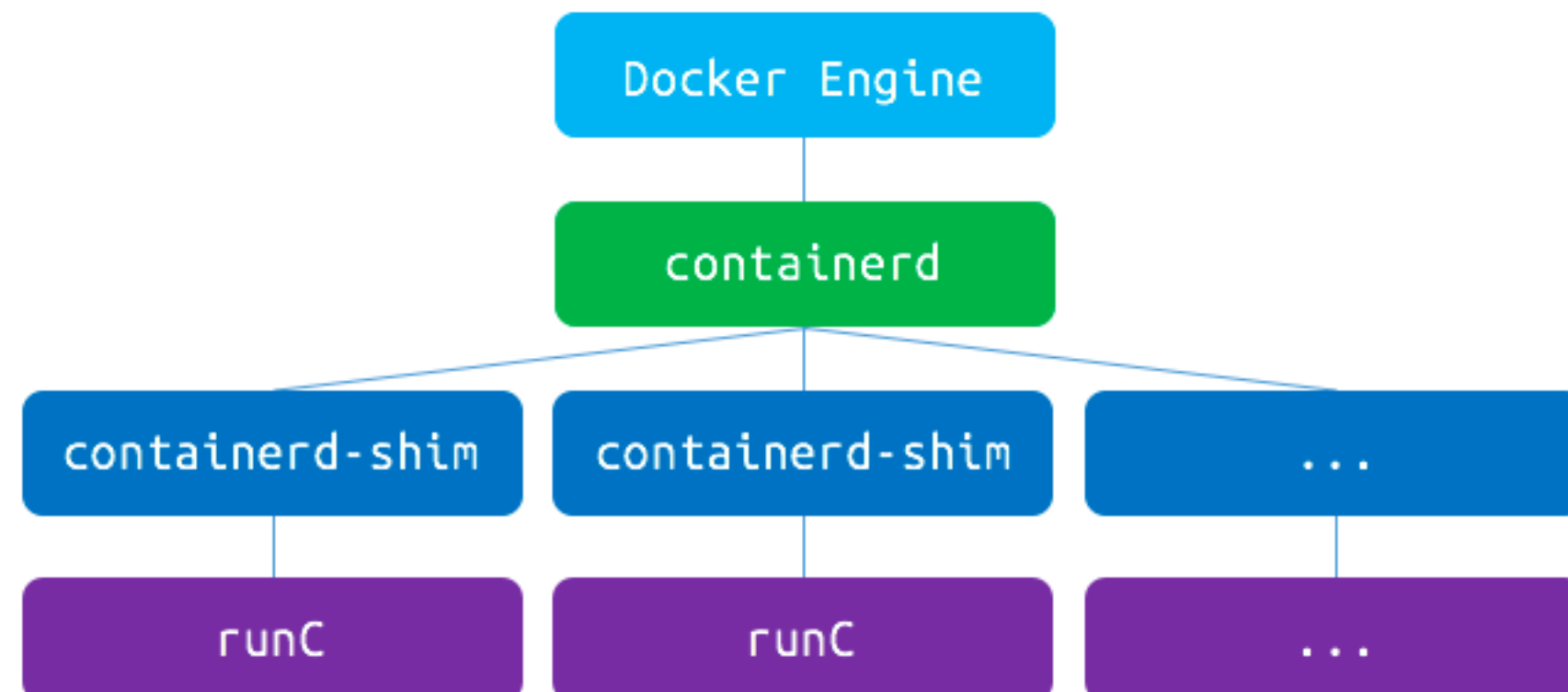
그 외 장점

1. Dockerhub에 유명한 Docker image들이 많음 (like github)
2. 갑자기 python 2.73이 필요하면 python:2.73 docker image 받아서 쓰고 지우면 됨
3. 다른 os에서도 잘 돌아갈까? 받아서 테스트 하면 됨
4. git과 같이 image의 변경 이력을 저장
5. 이미지 용량

Docker

도커를 이루는 기술

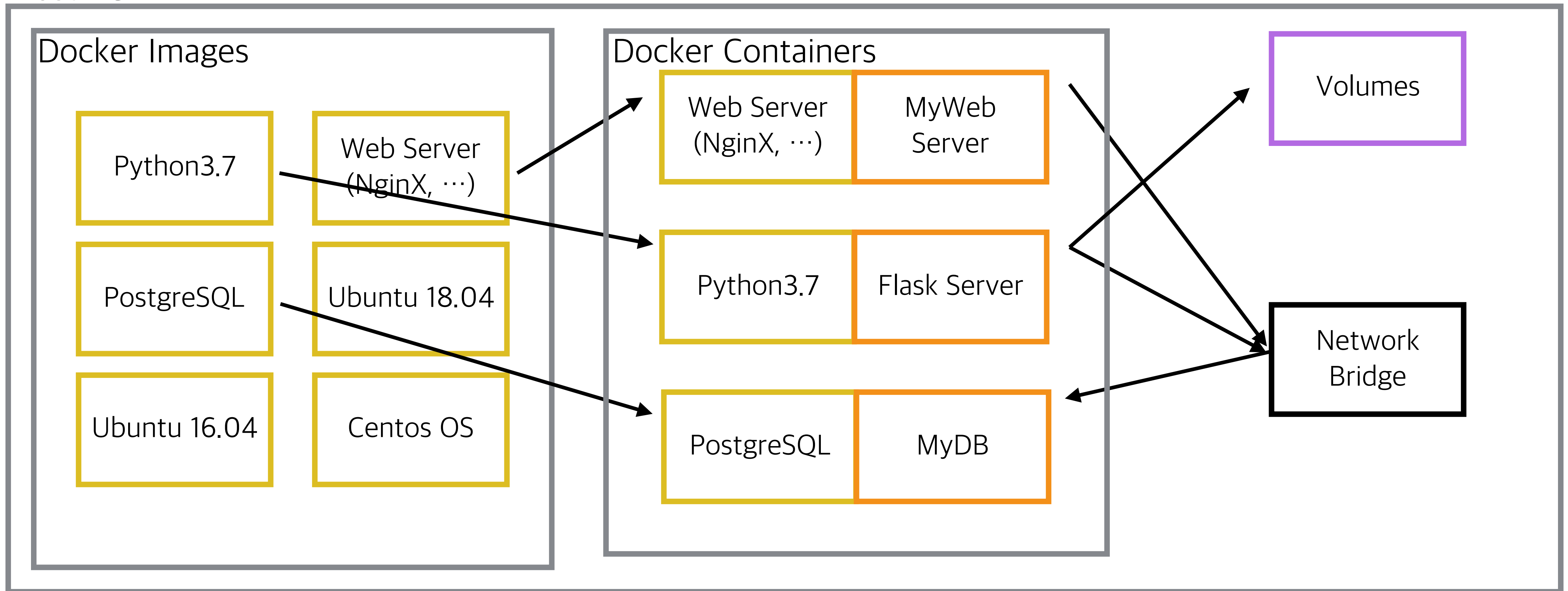
1. Lxc, runC등의 linux container 구현체를 이용
2. linux의 cgroups, namespaces등을 이용
3. namespaces는 독립적인 공간을 할당해주는 기술
4. cgroups는 자원의 사용에 대한 제어를 해주는 기술
5. 도커란 기존의 기술을 잘 조립해서 패키징한 툴



Docker

도커의 모습

Host PC



Docker

이미지와 컨테이너

1. 도커 이미지는 잘 패키징한 형태
2. 도커 컨테이너는 도커 이미지를 실행한 형태

*** 이 말은 즉 배포할 최종 결과는 도커 이미지이고, 배포하여 실행된 형태를 도커 컨테이너라는 말**

Docker

도커 이미지

1. 유명한 오픈소스 제품들은 대부분 잘 패키징한 도커 이미지가 있음
2. 예를들면 우분투 도커 이미지는 https://hub.docker.com/_/ubuntu
3. 파이썬 도커 이미지는 https://hub.docker.com/_/python 에 있음
4. 보통 base image에서 시작하여 필요한 환경을 셋팅하는 방식을 이용

Docker

도커 이미지 만드는 방법

1. 보통 base image에서 시작하여 필요한 환경을 셋팅하는 방식을 이용
 1. 도커 이미지를 실행하여 (컨테이너화) 필요한 환경을 만든뒤 저장하는 방법 (commit)
 2. Dockerfile을 작성해서 저장하는 방법

```
Dockerfile
```

```
FROM python:3.6-onbuild
```

```
RUN mv /usr/src/app /
```

```
WORKDIR /app
```

2. 보통 Dockerfile을 작성해서 이미지를 build하는 것이 일반적임

Q & A