

DOCKER SEOUL MEETUP

Docker란 무엇인가? : Docker 기본 사용법

이재홍

<http://pyrasis.com>

Docker Seoul Meetup #1 / August 23, 2014

기존의 서버 구축

- OS 설치
- 웹 서버 설치 및 설정
- DB 설치 및 설정
- 소스 복사
- 사람이 일일이 설정

Immutable Infrastructure

- 호스트OS와 서비스 운영 환경 분리
- 한 번 설정한 운영 환경은 변경하지 않는다(Immutable).
- 서비스 운영 환경을 통째로 교체한다.
- 마치 클라우드 플랫폼처럼...



서버



서버

서비스 운영 환경 이미지 생성

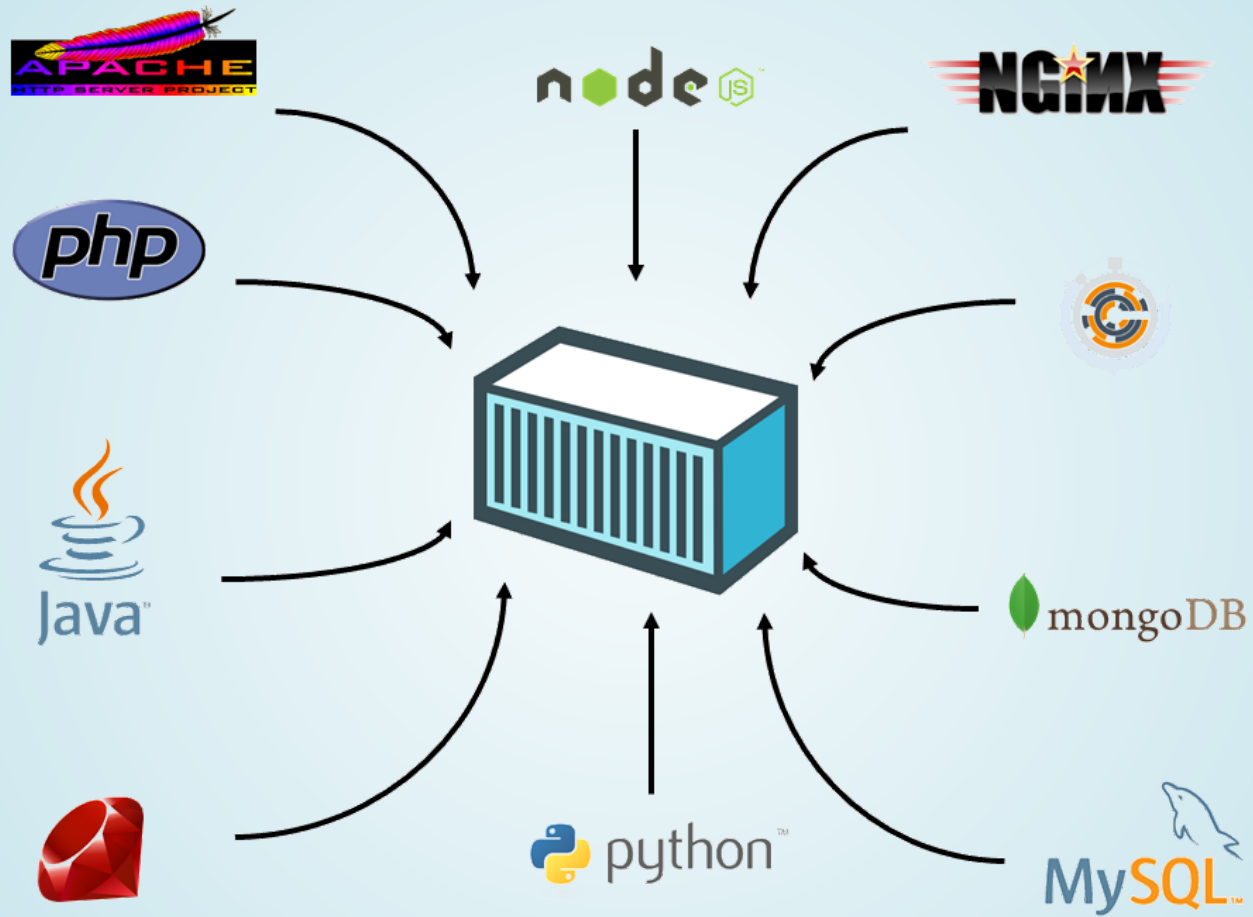


...

배포 및 실행



docker

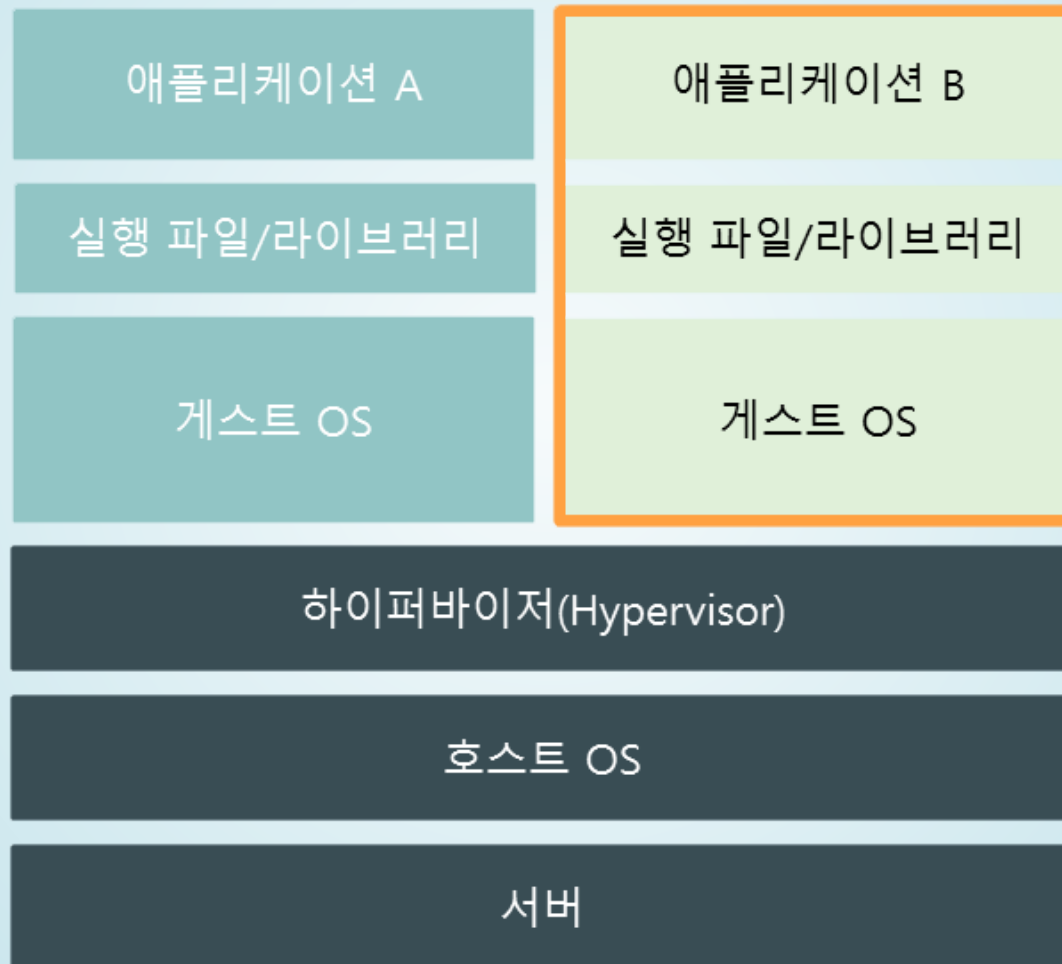


Virtual Machine

vs.

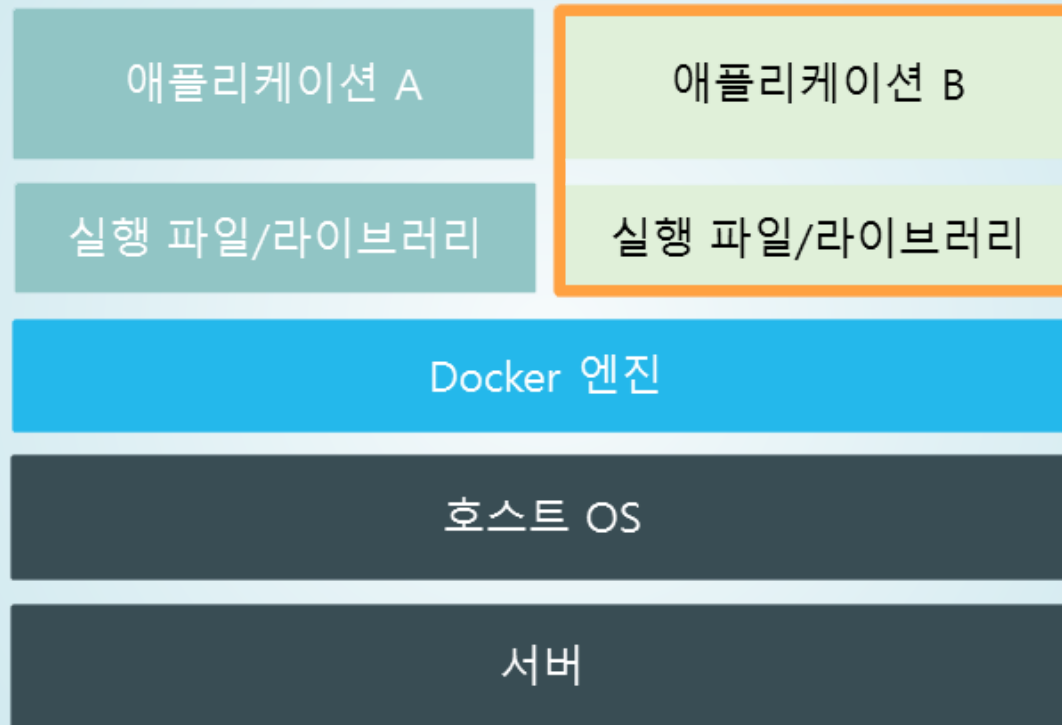
Docker

Virtual Machine



<http://www.docker.com/whatisdocker/>

Docker

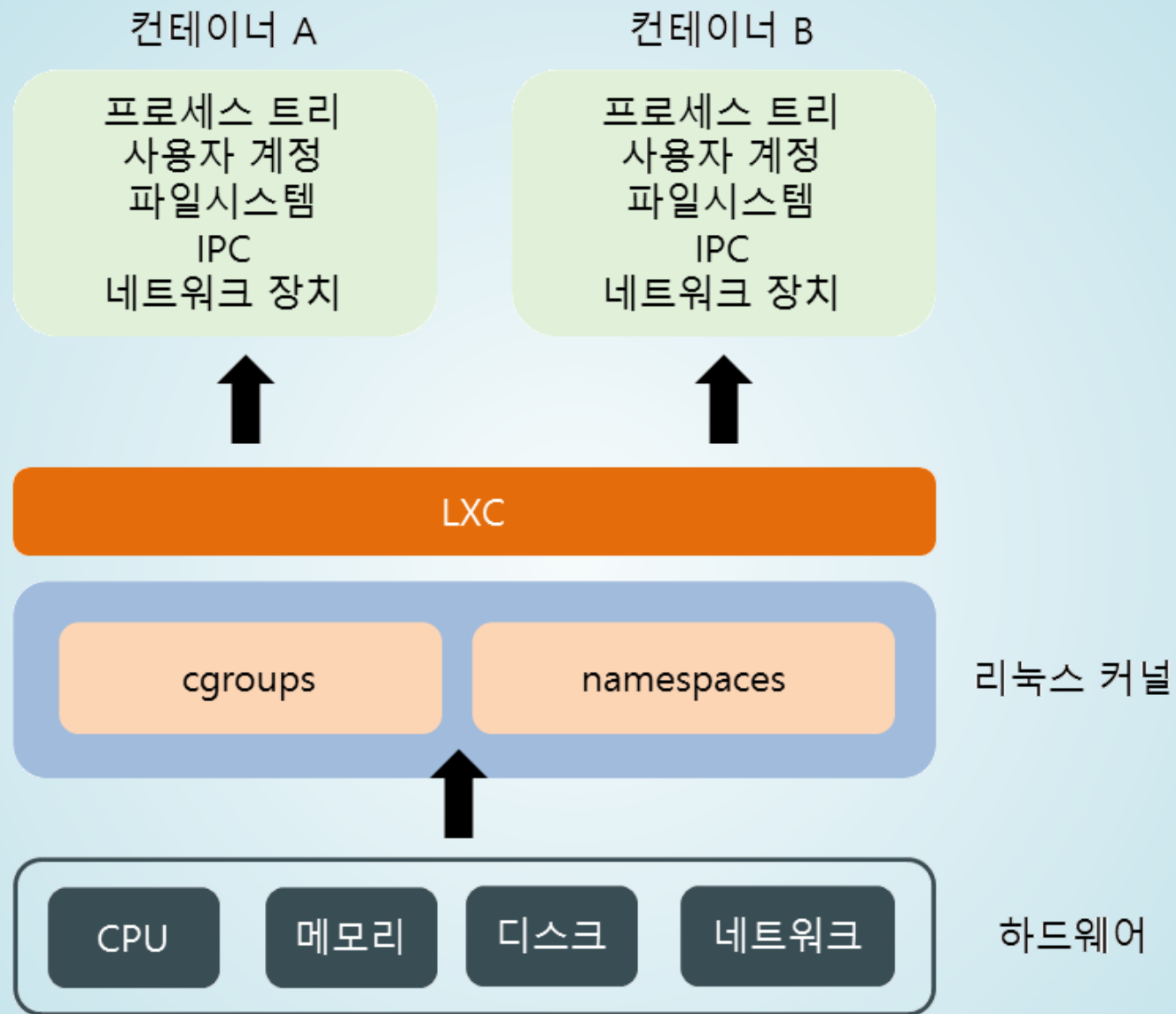


<http://www.docker.com/whatisdocker/>

	성능 측정 도구	호스트	Docker
CPU	sysbench	1	0.9945
메모리 쓰기	sysbench	1	0.9826
메모리 읽기	sysbench	1	1.0025
디스크 I/O	dd	1	0.9811
네트워크	iperf	1	0.9626

Ubuntu 14.04, Docker 1.1.2

Linux Container



Docker

=

cgroups, namespaces...

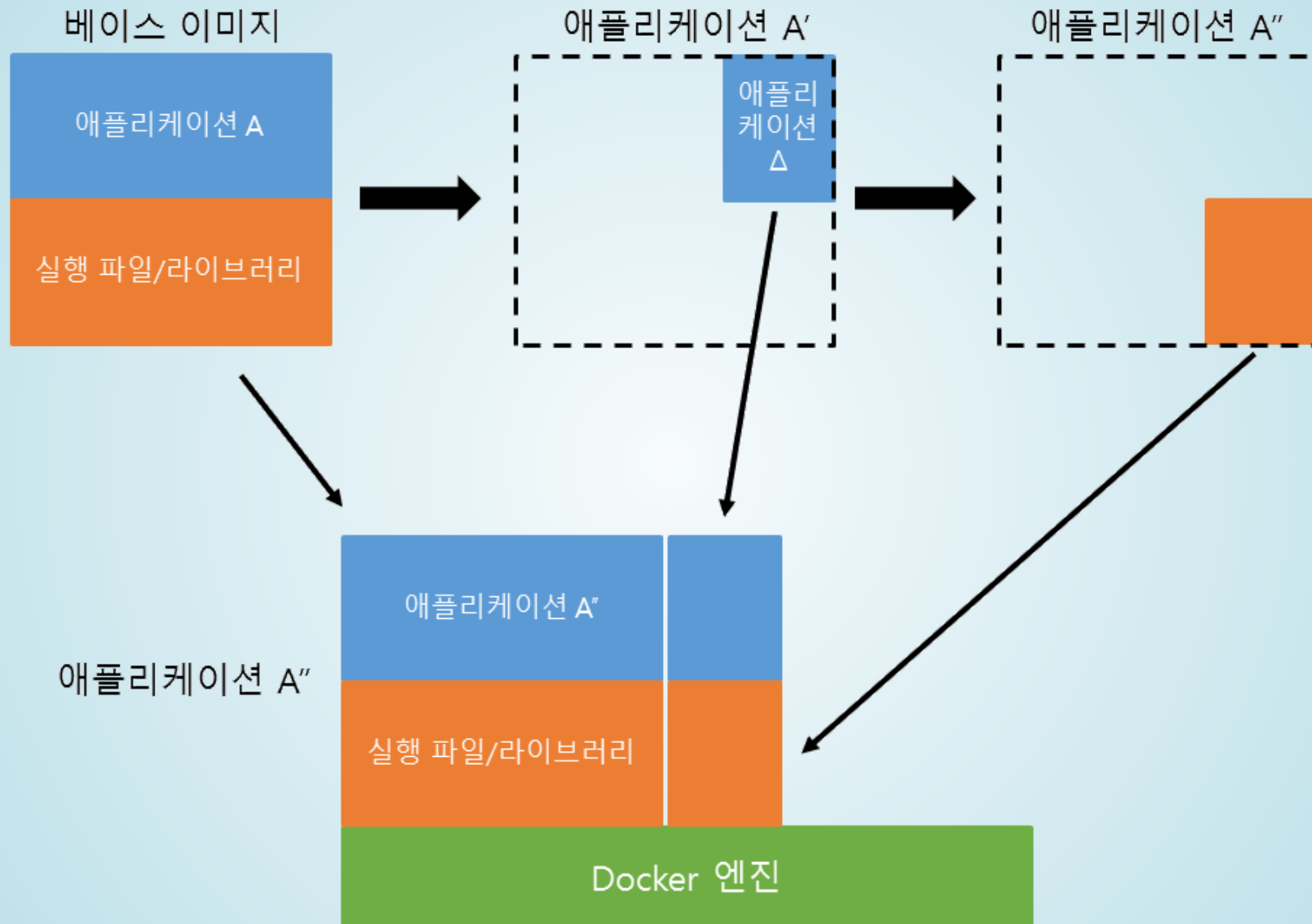
+

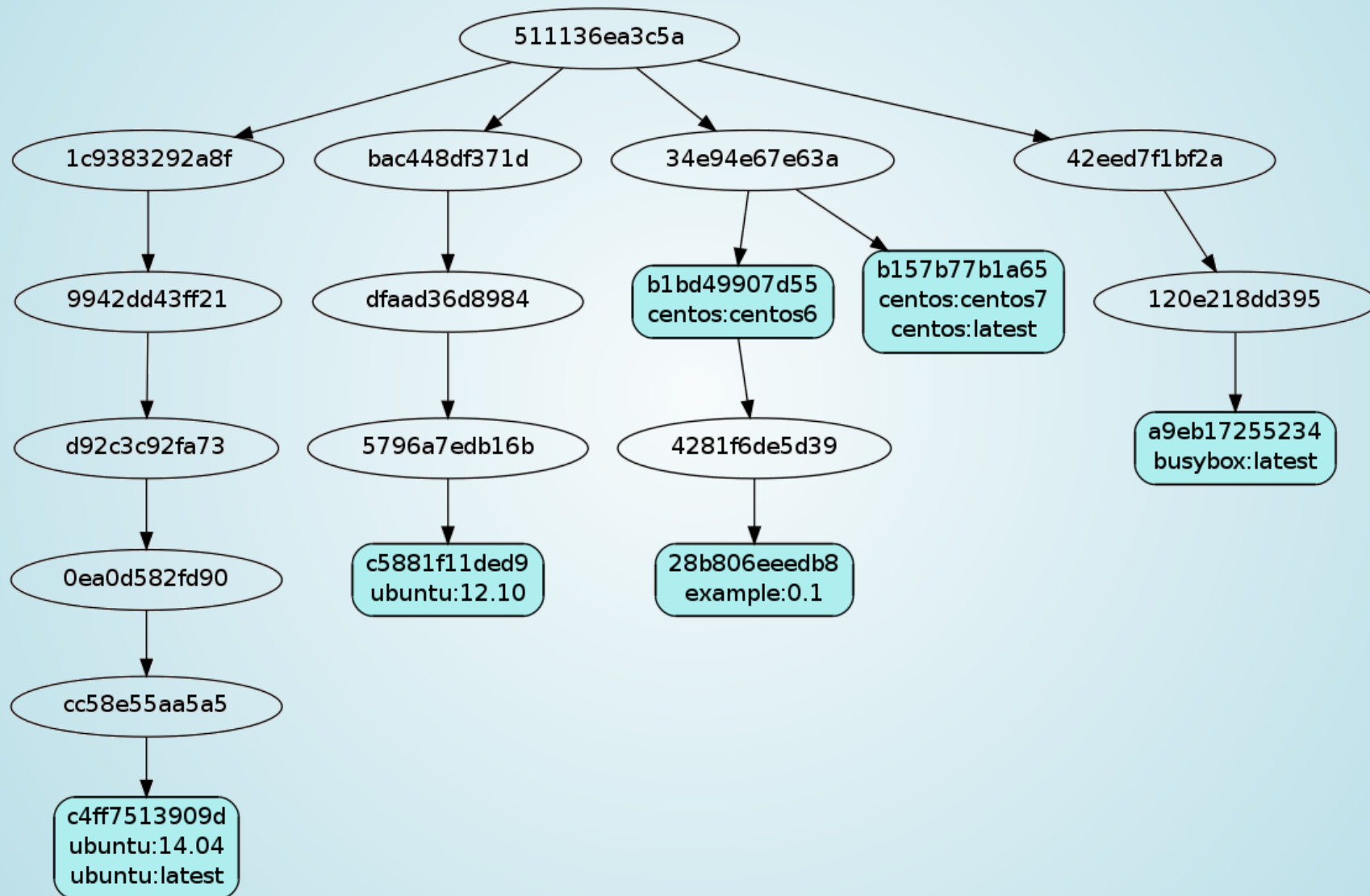
Docker Hub

+

α

Docker 이미지와 컨테이너





Docker 이미지

실행 파일, 라이브러리, 소스 등을 묶은 이미지 파일

예) OS의 실행 파일

Docker 컨테이너

이미지를 실행한 상태

예) OS의 프로세스

설치

```
$ sudo wget -qO- https://get.docker.io/ | sh
```

Ubuntu

```
$ sudo apt-get update
$ sudo apt-get install docker.io
$ sudo ln -sf W
    /usr/bin/docker.io /usr/local/bin/docker
```

CentOS

CentOS 6.5

```
$ sudo yum install W  
    http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm  
$ sudo yum install docker-io
```

CentOS 7

```
$ sudo yum install docker
```

Mac OS X

<https://github.com/boot2docker/osx-installer/releases>

Windows

<https://github.com/boot2docker/windows-installer/releases>

Mac OS X와 Windows용 Docker는

- Boot2Docker
- VirtualBox가 내장
- 가상 머신으로 리눅스를 실행한 것

기본 실습

매번 sudo를 붙이지 않으려면...

```
$ sudo groupadd docker  
$ sudo gpasswd -a ${USER} docker  
$ sudo service docker restart
```


이미지 검색

```
$ docker search ubuntu
```

이미지 받기

```
$ docker pull ubuntu:latest
```

이미지 목록 출력하기

```
$ docker images
```

컨테이너 생성하기

```
$ docker run -i -t --name hello W  
ubuntu /bin/bash
```

컨테이너 목록 확인하기

```
$ docker ps -a
```

컨테이너 시작하기

```
$ docker start hello
```

컨테이너 재시작하기

```
$ docker restart hello
```

컨테이너에 접속하기

```
$ docker attach hello
```

컨테이너 정지하기

```
$ docker stop hello
```

컨테이너 삭제하기

```
$ docker rm hello
```

이미지 삭제하기

```
$ docker rmi ubuntu:latest
```

모든 컨테이너 삭제하기

```
$ docker rm `docker ps -aq`
```

이미지 생성하기

Dockerfile

```
FROM ubuntu:14.04
MAINTAINER Foo Bar <foo@bar.com>

RUN apt-get update
RUN apt-get install -y nginx
RUN echo "Wndaemon off;" >> /etc/nginx/nginx.conf
RUN chown -R www-data:www-data /var/lib/nginx

VOLUME ["/data", "/etc/nginx/site-enabled", "/var/log/nginx"]

WORKDIR /etc/nginx

CMD ["nginx"]

EXPOSE 80
EXPOSE 443
```

- FROM: 어떤 이미지를 기반으로 할지 설정
- MAINTAINER: 이미지 작성자 정보
- RUN: 이미지에서 스크립트나 명령 실행
- CMD: 컨테이너가 시작되었을 때 스크립트나 명령 실행
- ENTRYPOINT: 컨테이너가 시작되었을 때 스크립트나 명령 실행(docker run에서 처리 방식이 다름)
- EXPOSE: 호스트와 연결할 포트 번호 설정
- ENV: 환경 변수 설정
- ADD, COPY: 이미지에 파일 추가
- VOLUME: 데이터를 호스트에 저장하도록 설정
- USER: 명령을 실행할 사용자 계정 설정
- WORKDIR: 명령을 실행할 디렉터리 설정
- ONBUILD: FROM으로 이미지가 사용될 때 실행할 명령 설정

이미지 생성하기

```
$ docker build --tag hello:0.1 .
```

컨테이너 생성하기

```
$ docker run --name hello-nginx -d -p 80:80 W  
-v /root/data:/data hello:0.1
```


기타 명령

이미지 히스토리 살펴보기

```
$ docker history hello:0.1
```

컨테이너에서 파일 꺼내기

```
$ docker cp W  
hello-nginx:/etc/nginx/nginx.conf ./
```

컨테이너의 변경 사항을 이미지로 저장하기

```
$ docker commit -a "Foo Bar <foo@bar.com>" W  
-m "add hello.txt" hello-nginx hello:0.2
```

컨테이너에서 변경된 파일 확인하기

```
$ docker diff hello-nginx
```

이미지와 컨테이너의 세부정보 확인하기

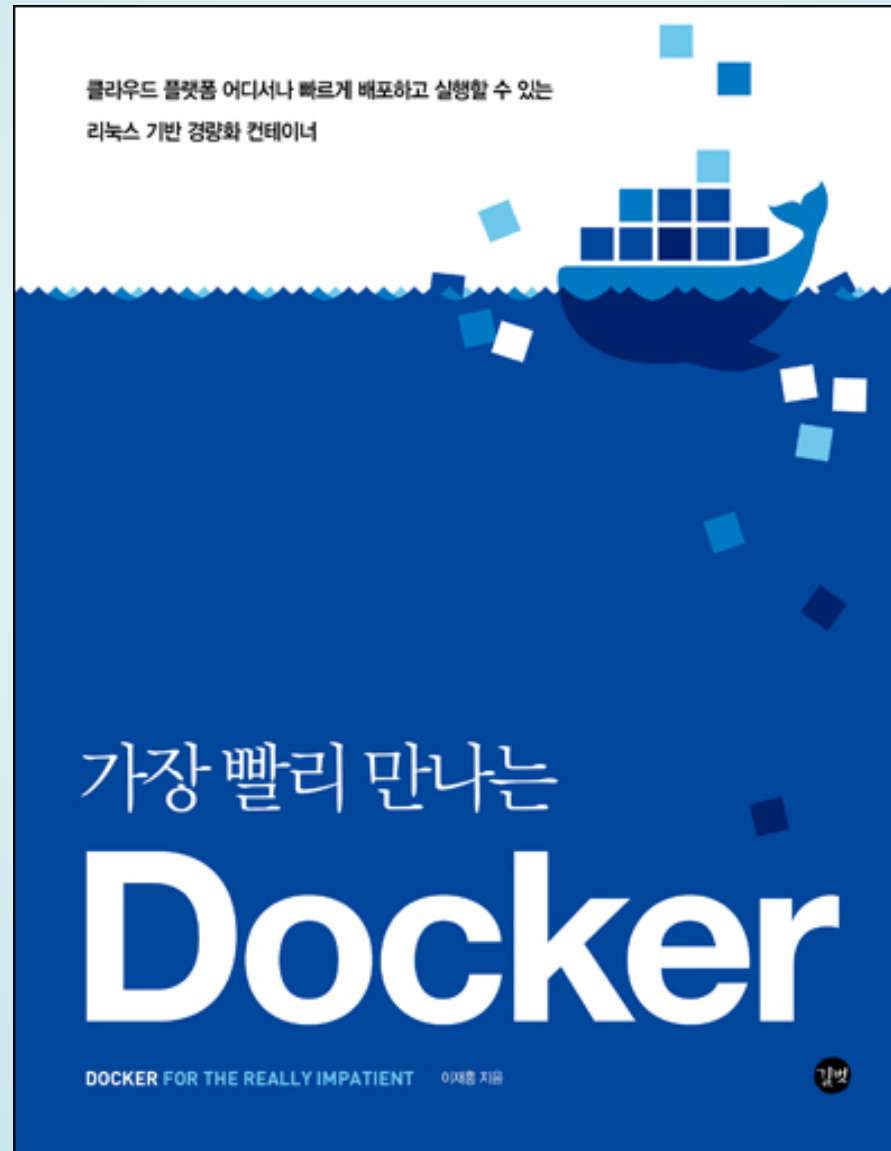
```
$ docker inspect hello-nginx
```

참고 문서

<http://pyrasis.com/Docker/Docker-HOWTO>

"가장 빨리 만나는 **Docker**" 출간 및 원고 공개

<http://www.pyrasis.com/private/2014/11/30/publish-docker-for-the-really-impatient-book>



"아마존 웹 서비스를 다루는 기술" 출간 및 원고 공개

<http://www.pyrasis.com/private/2014/09/30/publish-the-art-of-amazon-web-services-book>

