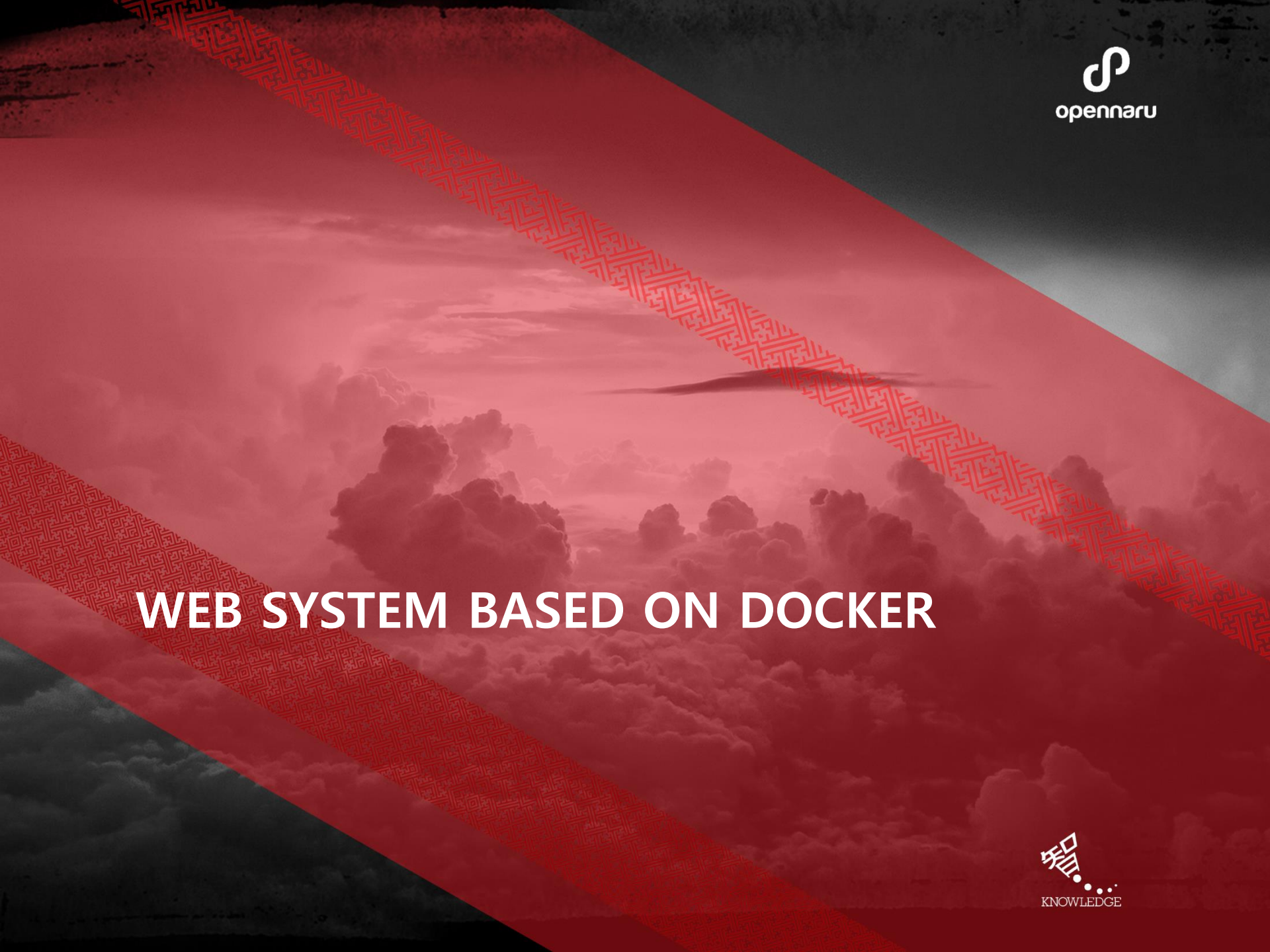

opennaru
WEB SYSTEM BASED ON DOCKER
KNOWLEDGE

Before Container

https://www.youtube.com/watch?v=G7GSYbY6iv8&index=2&list=PLaFPOkYzLL-8V-cqKy2BLEBn6tf0KjFFk

Opennaru, Inc. © 2016 | All Rights Reserved.

After Container

https://www.youtube.com/watch?v=jlOOjx0kZfk&index=1&list=PLaFPOkYzLL-8V-cqKy2BLEBn6tf0KjFFk

Opennaru, Inc. © 2016 | All Rights Reserved.


opennaru
Change brings opportunity.
Nido R. Qubein

Container 시대를 향하여

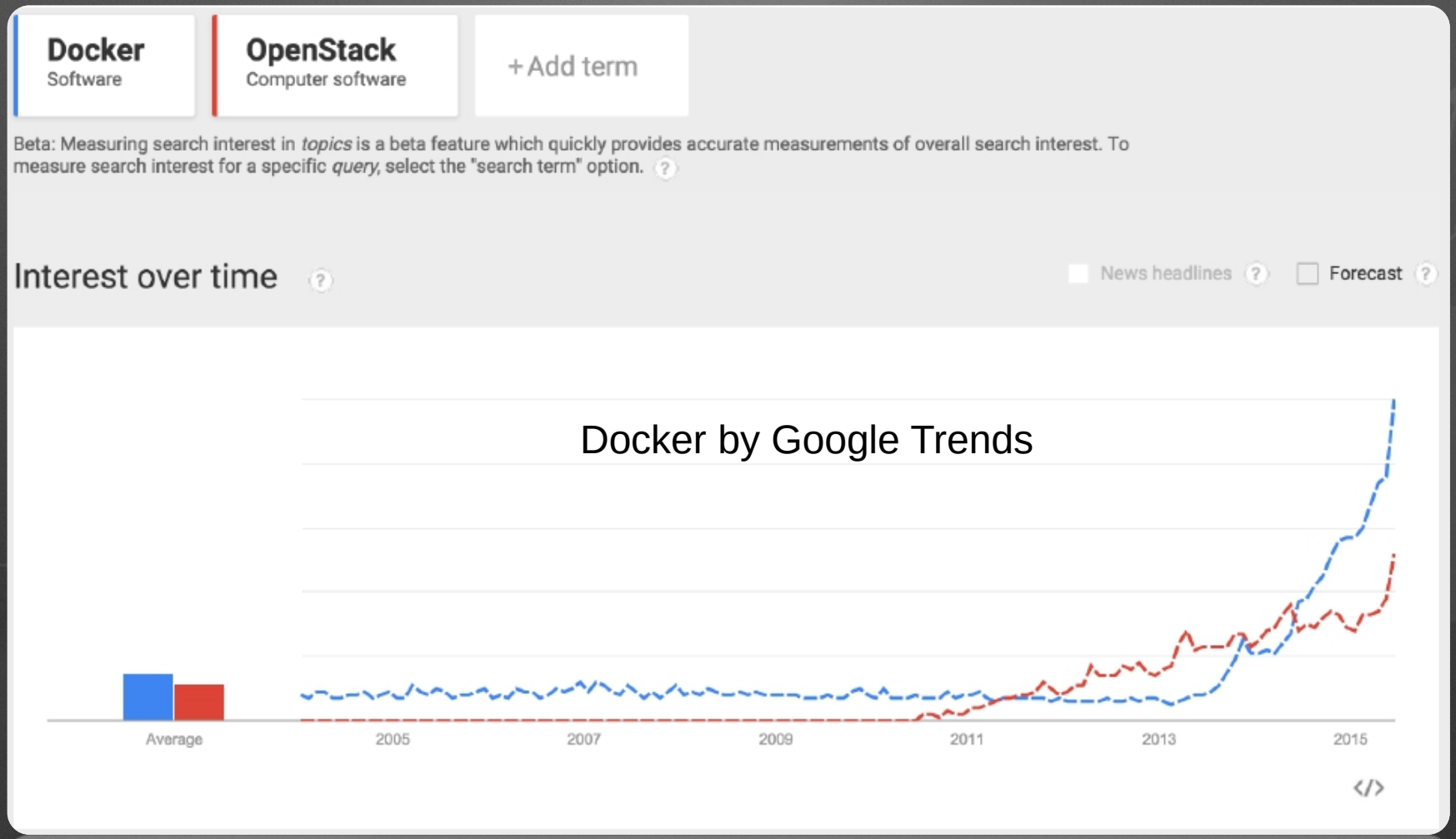
opennaru

1. 컨테이너란 무엇인가? 가상화와의 차이는 무엇인가?
   - ☐ 기존의 가상화와 컨테이너는 무엇이 다른지?
   - ☐ 컨테이너에는 어떤 구조로 되어 있는 건가?
   - ☐ Linux OS만 있으면 컨테이너를 바로 사용하나?
2. 컨테이너 적용시의 기대효과?
   - ☐ 컨테이너를 사용하면 무엇이 좋은 것인가?
   - ☐ 컨테이너는 무조건 좋은 건가? 단점은 무엇인가?
   - ☐ 컨테이너를 적용하기에 좋은 시스템과 그렇지 않은 시스템은?
3. 컨테이너를 운영 환경에서 사용하려면? 최근 화제인 컨테이너 플랫폼이란?
   - ☐ 컨테이너기반 시스템을 구축 하는 방법은?
   - ☐ 컨테이너는 실제 운영환경에서 사용할 만큼 성숙된 기술인가?
   - ☐ 컨테이너 플랫폼은 무엇이 있으며 향후 발전 방향은?

- Confidential -

Opennaru, Inc. © 2016 | All Rights Reserved.

Docker by Google Trends



Opennaru, Inc. © 2016 | All Rights Reserved.

Docker 란?

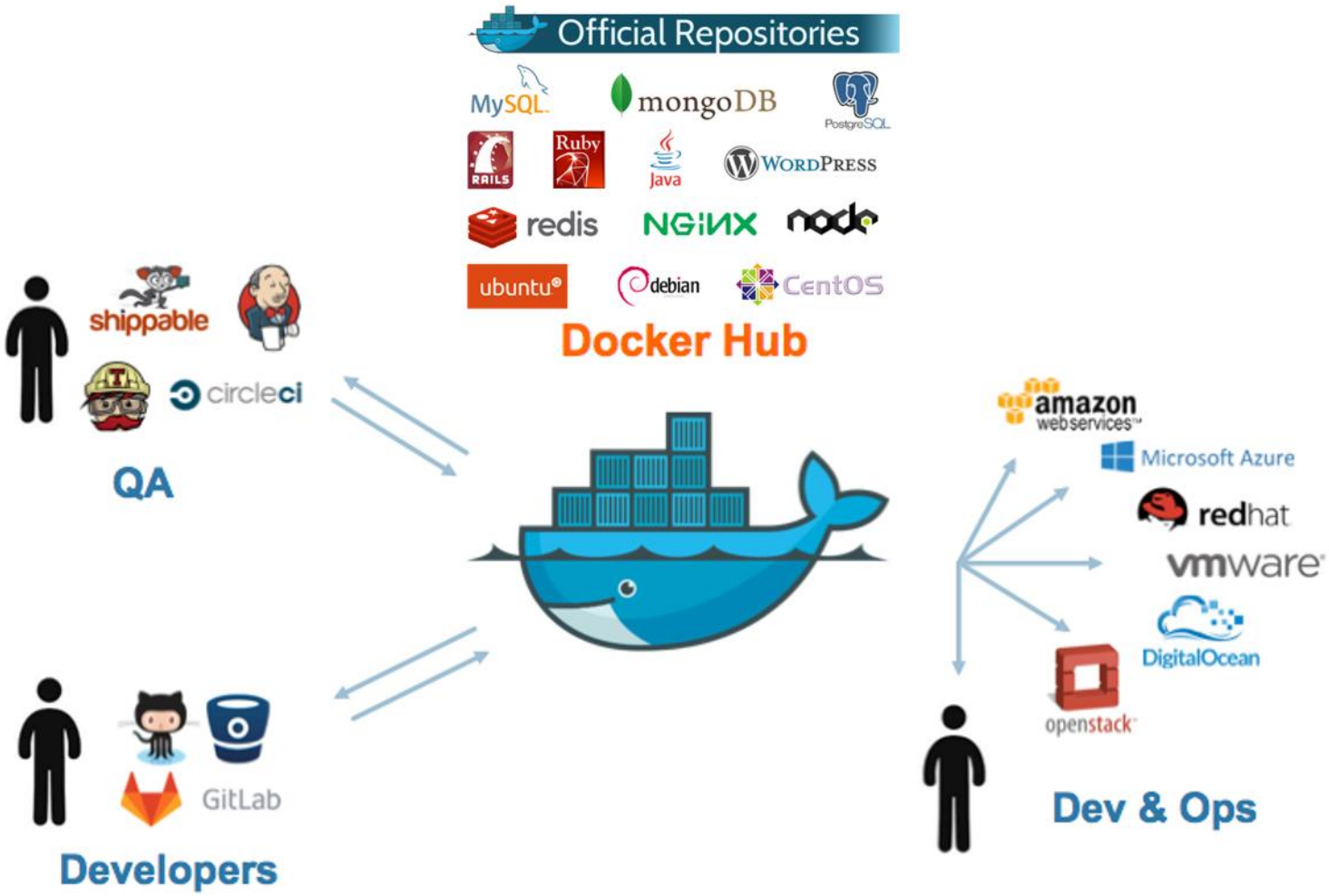
- 고래 등에 올려진 컨테이너 박스들 처럼
- 프로그램과 실행에 필요한 것들을 컨테이너에 Shipping,
- 컨테이너를 손쉽게 이동해서,
- 어디서나 간단하게 실행할 수 있는
- 도구와 환경을 제공하는 오픈 소스 플랫폼.

Build, Ship and Run
Any App, Anywhere

Opennaru, Inc. © 2016 | All Rights Reserved.

Docker hub

- Docker는 Linux 커널을 사용하고 있는 환경에 있다면 어디서나 작동하기 때문에 플랫폼을 의식하지 않고 사용 (Linux 만)



Opennaru, Inc. © 2016 | All Rights Reserved.

Docker vs. 가상화

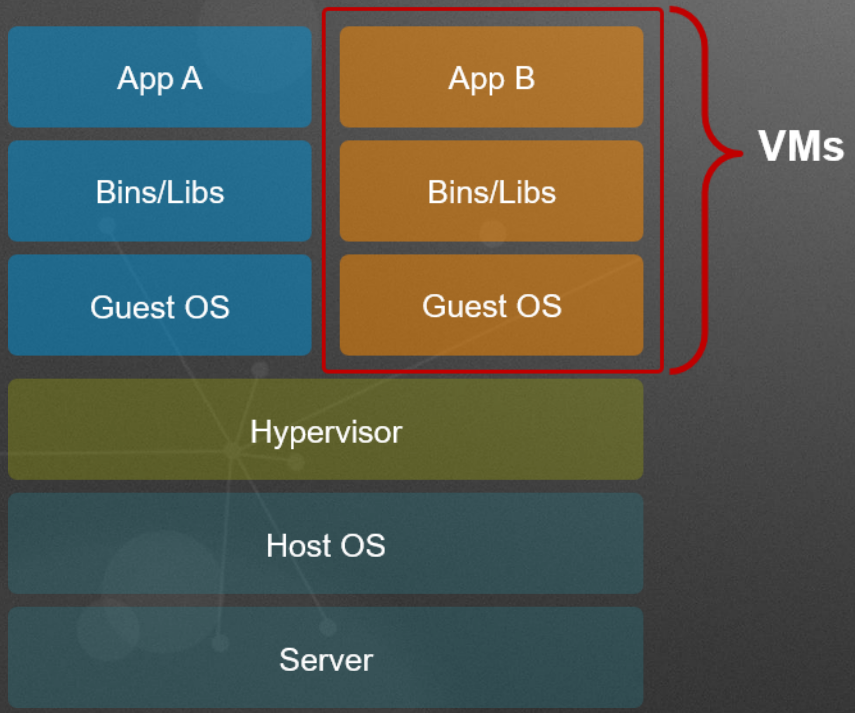
opennaru

	컨테이너 형 가상화 (Docker)	하이퍼 바이저 형 가상화 (VMWare ESXi)	호스트 형 가상화 (Linux KVM)
가상 머신	OS를 호스트 OS와 공유하기 때문에 VM 마다 OS 설치를 할 필요는 없다	VM마다 OS 설치	VM마다 OS 설치
지원 OS	• Linux • Windows	• Windows, Linux • 일부 Unix도 지원	• Windows, Linux • 일부 Unix도 지원
부팅 시간	OS 설치 불필요하기 때문에 사용 시작까지의 시간이 매우 짧음	초기 구축 시에는 네트워크 OS 설치 등의 작업이 발생하기 때문에 이용 개시까지의 시간이 많이 소요	초기 구축 시에는 네트워크 OS 설치 등의 작업이 발생하기 때문에 이용 개시까지의 시간이 많이 소요
네트워크	호스트 측에 작성된 Docker 전용 NIC와 통신	• 네트워크의 생성이 가능 • VM에 임의의 숫자 vNIC를 부여 가능	• 네트워크의 생성이 가능 • VM에 임의의 숫자 vNIC를 부여 가능
자원	표준에서는 HDD 자원을 지정할 수 없다. CPU, 메모리에 대한 자원 할당 지정 가능	CPU, 메모리, HDD의 자원 할당을 지정	CPU, 메모리, HDD의 자원 할당을 지정
오버 헤드	컨테이너는 호스트 OS에서 보면 하나의 프로세스이며, 오버 헤드는 거의 없음	VM에서 기기까지의 액세스 경로를 하이퍼바이저 뿐이므로 호스트 형 가상화에 비해 오버 헤드가 적은	VM에서 기기까지의 액세스 경로가 다른 가상화 기술에 비해 길기 때문에 비교했을 경우에는 가장 오버 헤드가 높음

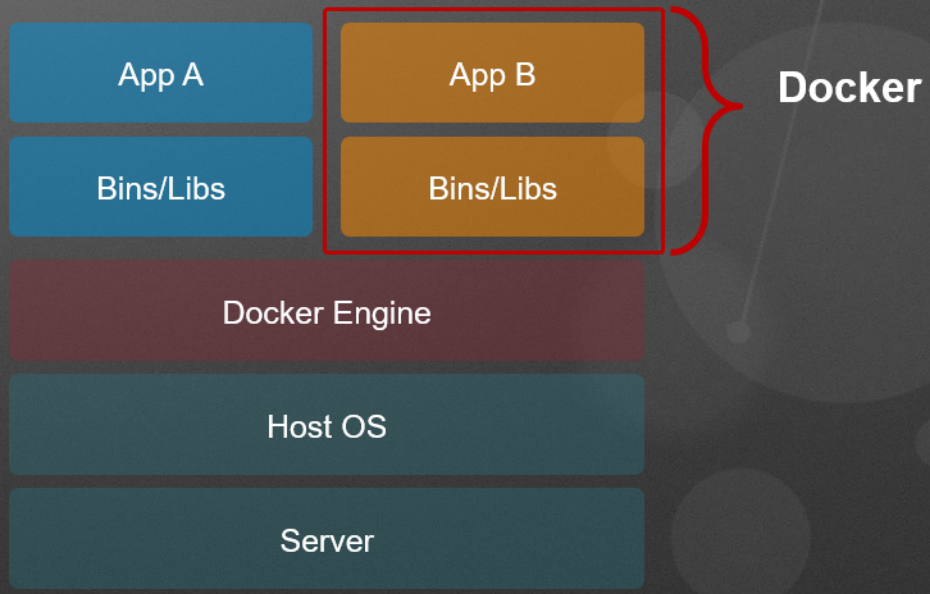
Opennaru, Inc. © 2016 | All Rights Reserved.

Containers vs. VMs

opennaru

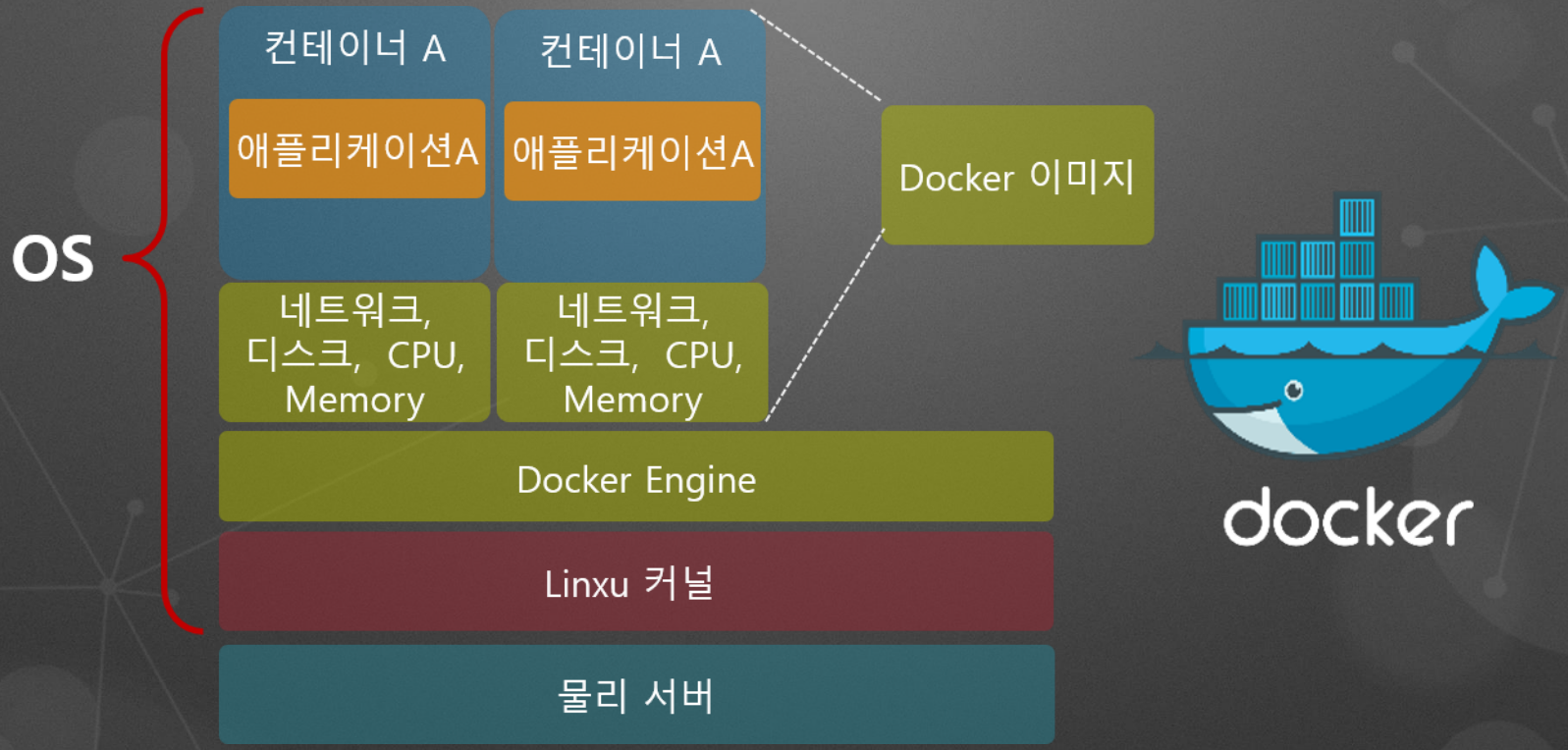


isolated process in userspace on the host operating system, sharing the kernel with other containers



Opennaru, Inc. © 2016 | All Rights Reserved.

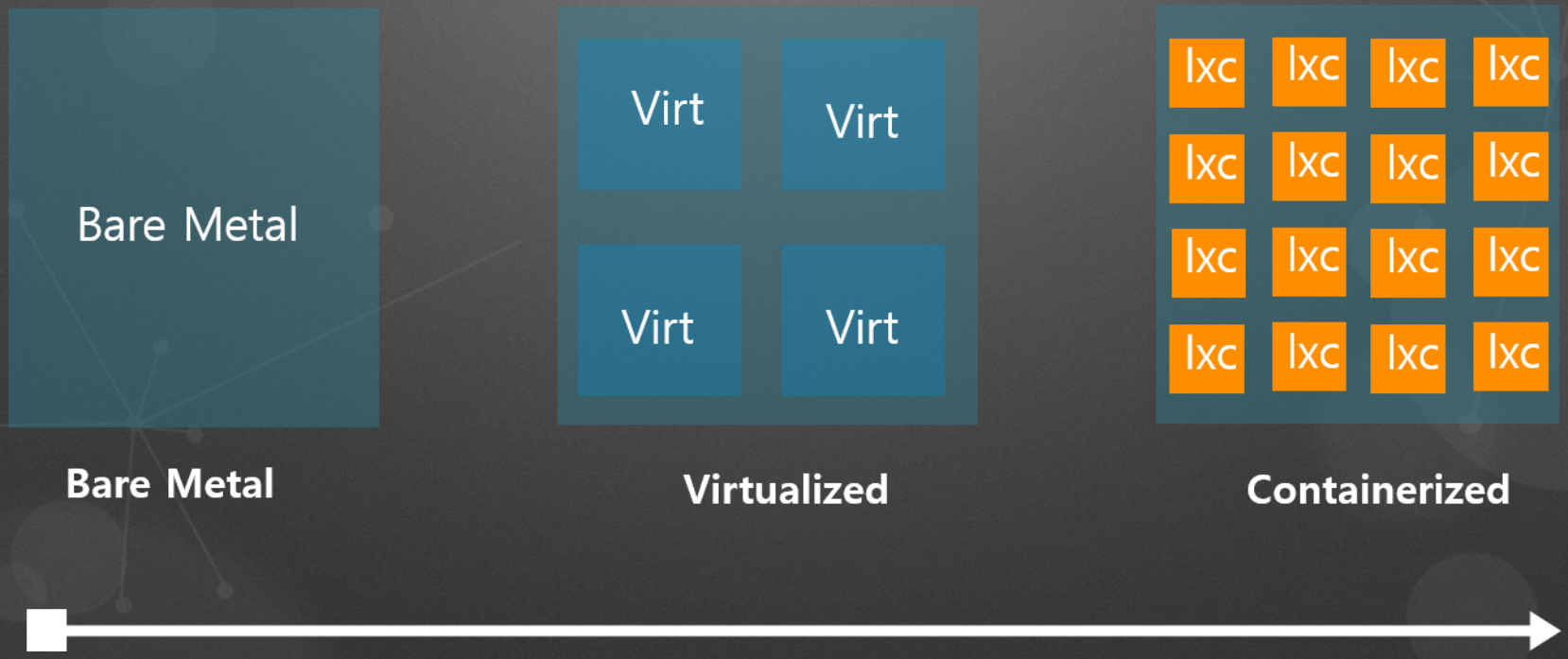
Docker Image



Opennaru, Inc. © 2016 | All Rights Reserved.

Evolution of Infrastructure Architectures

opennaru

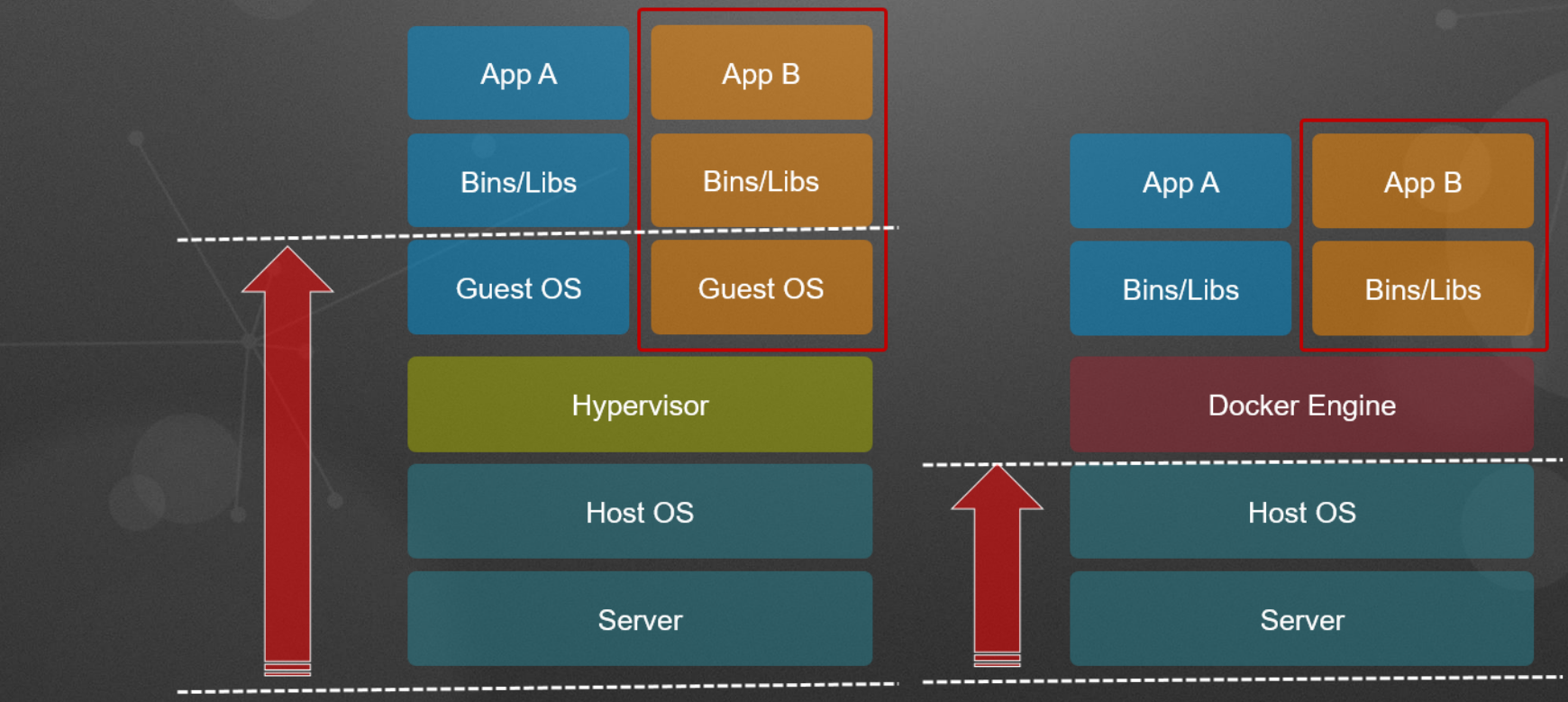


Opennaru, Inc. © 2016 | All Rights Reserved.

시작 시간 - Containers vs. VMs

opennaru

- 하드웨어 가상화는 CPU, 메모리, 하드 디스크 등의 하드웨어를 가상화하고 있기 때문에 하드웨어 나 OS 부팅해야 부팅에 분 단위 시간 소요
- 컨테이너 형 가상화에서는 컨테이너 부팅 시 OS는 이미 시작하고 프로세스의 시작 만 할 초 단위 시간

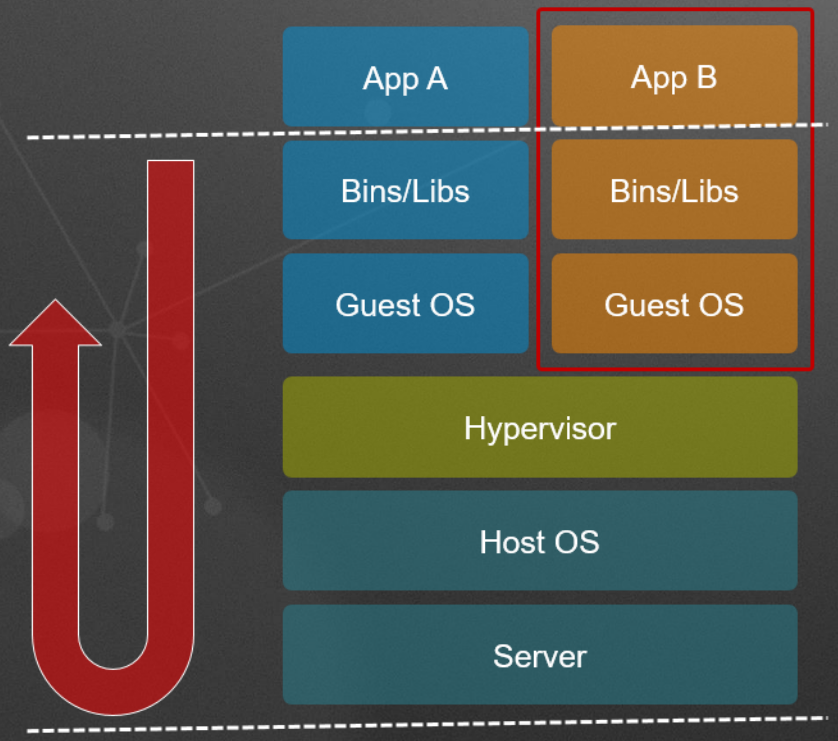


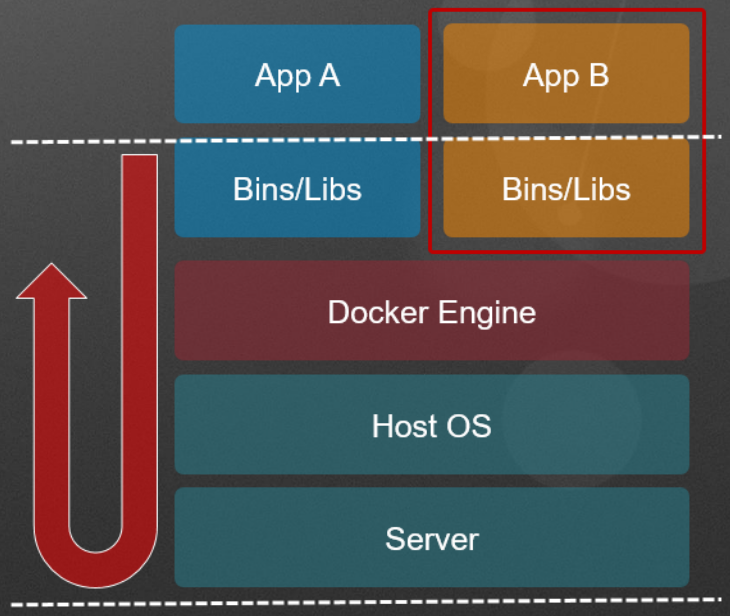
Opennaru, Inc. © 2016 | All Rights Reserved.

오버헤드 - Containers vs. VMs

opennaru

- OS에서 응용 프로그램을 작동하는 경우, 하드웨어 가상화에서는 가상화 된 하드웨어 및 하이퍼바이저를 통해 처리하기 때문에 물리적 시스템보다 처리에 부가적인 시간(오버 헤드)가 필요
- 컨테이너 형 가상화 커널을 공유하고 개별 프로세스가 작업을 하는 것과 같은 정도의 시간 밖에 걸리지 않기 때문에 대부분 오버 헤드가 없음



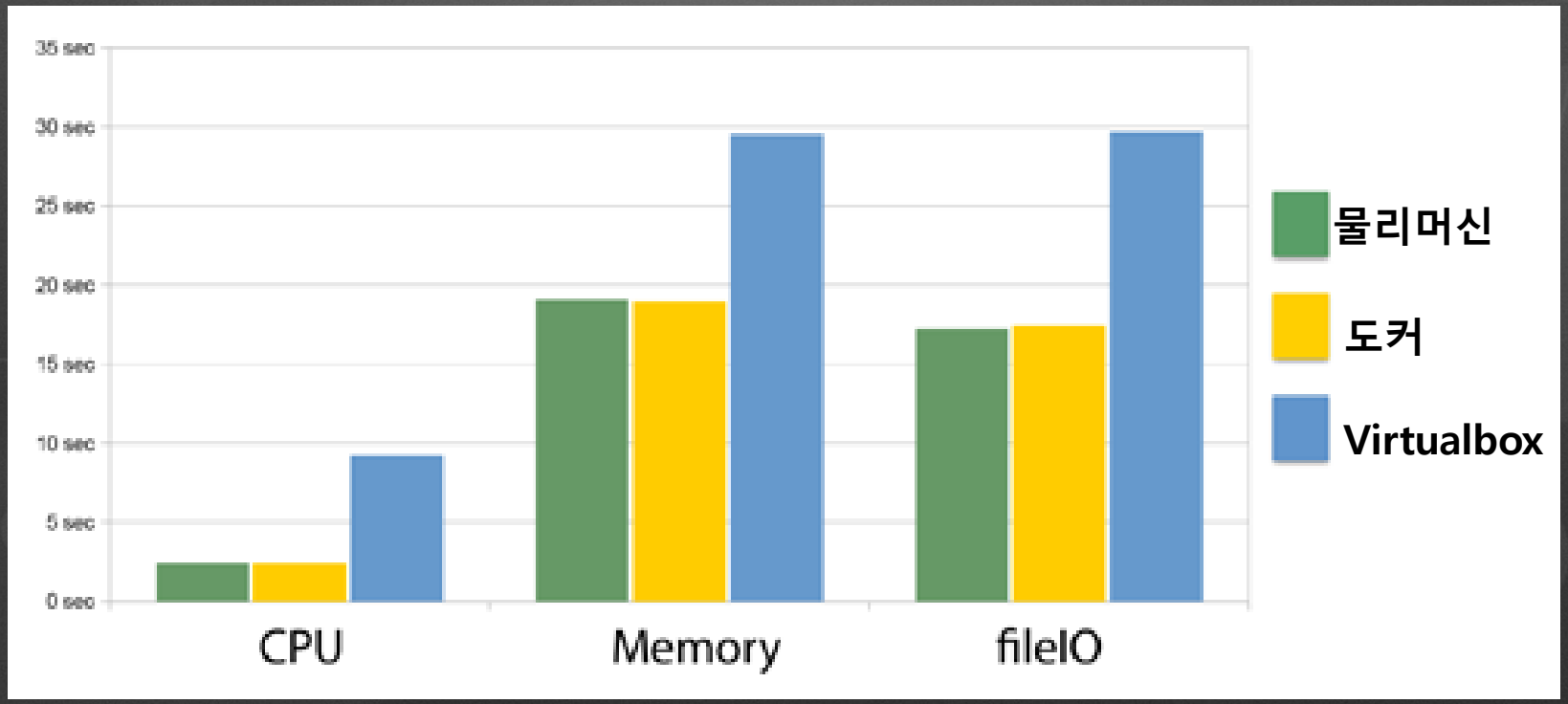


Opennaru, Inc. © 2016 | All Rights Reserved.

성능 - Containers vs. VMs

opennaru

- "sysbench"라는 벤치 마크 도구를 사용하여 성능 측정
- 물리적 시스템과 컨테이너 형 가상화 성능은 **모든 항목에서 거의 같은 결과**
- 하드웨어 가상화는 **메모리, 파일 IO는 약 2 배, CPU는 약 5 배의 시간**
- **물리 머신과 비교해도 성능 저하가 거의 없음**



Opennaru, Inc. © 2016 | All Rights Reserved.

History of Container

opennaru

2000:
JAILS ADDED
TO FREEBSD

2001:
LINUX -VSERVER
PROJECT

2003:
SELINUX ADDED TO
LINUX MAINLINE

2005:
FULL RELEASE OF
SOLARIS ZONES

2006:
GENERIC PROCESS
CONTAINERS

2007:
GPC RENAMED
CONTROL GROUPS

2008:
LINUX CONTAINER
PROJECT (LXC)

2008:
KERNEL AND USER
NAMESPACES

2013:
DOTCLOUD BECOMES
DOCKER

2013:
RED HAT ENTERPRISE
LINUX

2013:
DOT CLOUD PYCON
LIGHTNING TALK

2014:
GOOGLE KUBERNETES

2015:
RHEL ATOMIC HOST

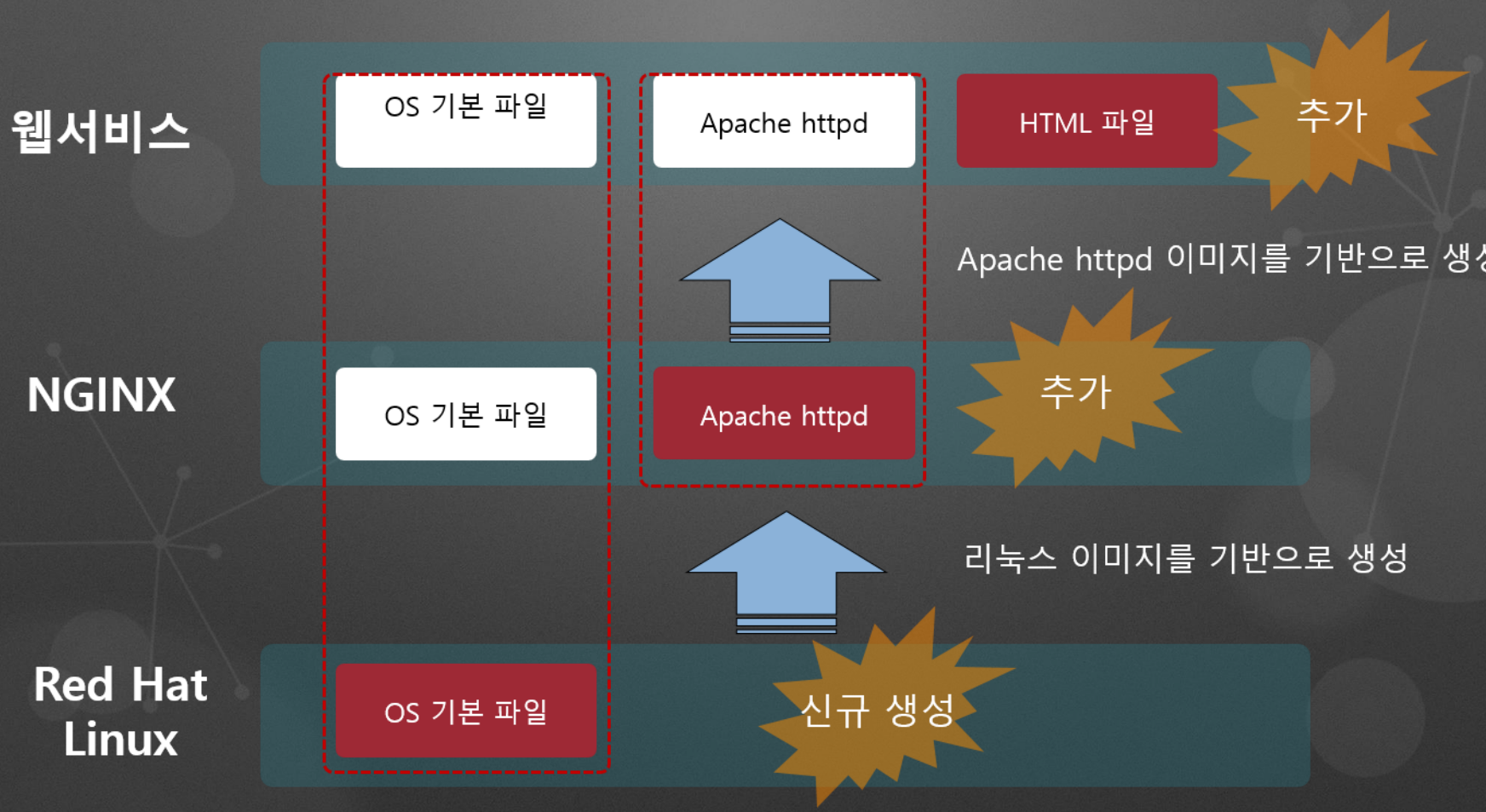
2015:
RHT CONTAINER
PLATFORMS

2015:
STANDARDS VIA OCI
AND CNCF

2000 2005 2010 2015

Opennaru, Inc. © 2016 | All Rights Reserved.

Docker 이미지 구조 예시



Opennaru, Inc. © 2016 | All Rights Reserved.

Google에서는 모든 것이 컨테이너 그리고 움직이고 있다

opennaru

Gmail, 검색, 맵, ...

MapReduce, 배치, ...

GFS, Colossus, ...

Google Cloud Platform도!

VM이 컨테이너로 움직이고 있다

매주 20억 개 이상의 컨테이너를 기동 하고 있다

Google

- Confidential -

Opennaru, Inc. © 2016 | All Rights Reserved.

Docker의 등장에 따른 컨테이너의 확산

opennaru

- 이용하기 위한 허들이 매우 낮다
  - 휴대용 이미지 포맷
  - 레지스트리에 의한 공유
  - 간단하게 취급할 수 있는 커멘드 라인 툴
  - 가벼운 런타임
- 환경을 구축하는 것이 비약적으로 편리해짐

```
$ docker run jenkins
```


- Confidential -

Opennaru, Inc. © 2016 | All Rights Reserved.

왜 컨테이너가 좋은 걸까?

opennaru

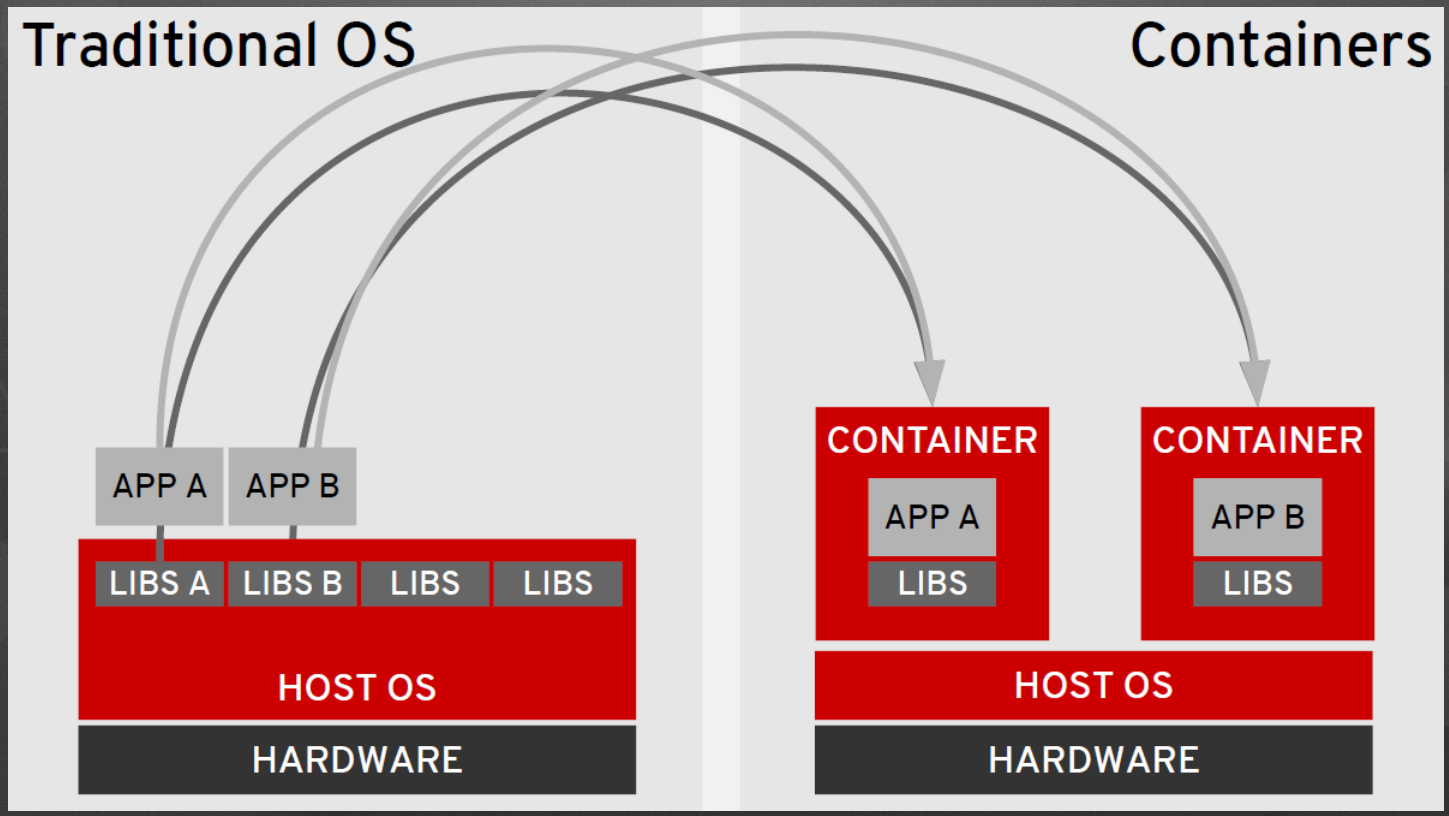
- 가볍다
  - 시작과 정지를 재빠르게 할 수 있다
  - 동시 실행 하는 컨테이너수가 많다.
  - 배포 사이즈가 작다
- 운영자 측면
  - 컨테이너 기술 자체의 학습도 용이(Docker)
  - 애플리케이션의 설치, 구성, 운영이 용이함
  - 컨테이너 공유 레포지토리 (Docker-Hub) 이용 가능
- 개발자 측면
  - 실행 환경의 차이를 고려할 필요가 없다

Opennaru, Inc. © 2016 | All Rights Reserved.

TRADITIONAL OS VS. CONTAINERS

opennaru

Packaged dependencies = faster boot times + greater portability



Opennaru, Inc. © 2016 | All Rights Reserved.

컨테이너 오케스트레이션 - Kubernates

opennaru

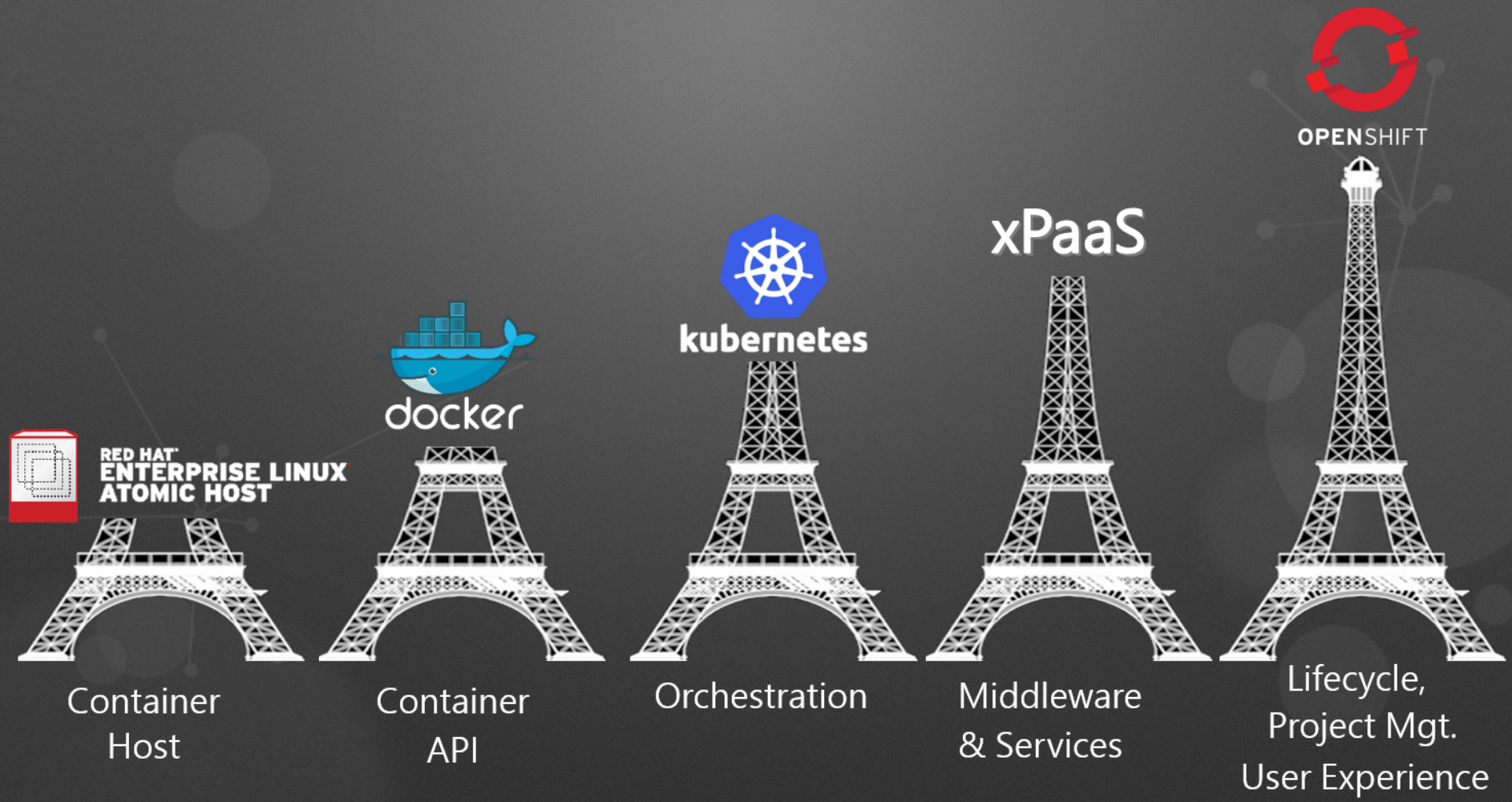
- 클라우드와 On-Promise 환경을 지원
- Google 경험으로 부터 설계
- Go로 쓰여져 있다
- open source
- 서버 관리보다는 애플리케이션 관리

- Confidential -

Opennaru, Inc. © 2016 | All Rights Reserved.

OVERVIEW: OPENSHIFT 3 Components

opennaru



Opennaru, Inc. © 2016 | All Rights Reserved.

OpenShift V3

opennaru

- PaaS에 필요한 기능 추가
  - 사용자 관리, 인증
  - 네트워크 분리
  - 소스에서 부터 배포 까지
- Docker, Kubernetes 등합

OPENSHIFT

- Web UI, 네트워크 관리, 사용자 관리
- Jenkins 연계
- 소스부터 서비스 구축 등의 서비스

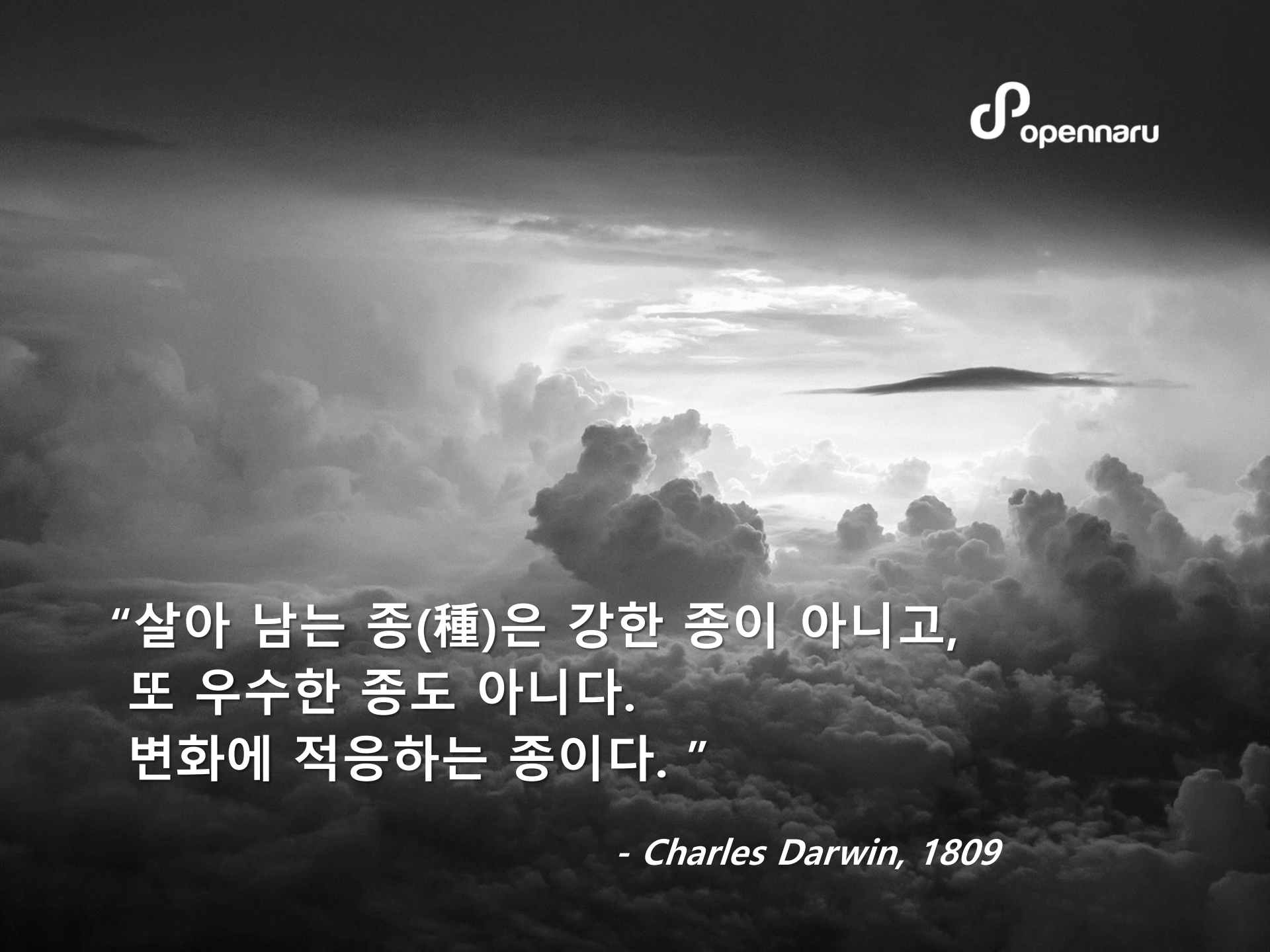
kubernetes

- 프록시, 로드밸런스 제공
- 컨테이너 라이프 사이클 관리
- 컨테이너를 조합해 서비스구성

docker

- 컨테이너 파일 포맷
- Linux 컨테이너 인터페이스

Opennaru, Inc. © 2016 | All Rights Reserved.


opennaru
“살아 남는 종(種)은 강한 종이 아니고,
또 우수한 종도 아니다.
변화에 적응하는 종이다. ”
- Charles Darwin, 1809

opennaru

감사합니다.

제품이나 서비스에 관한 문의

콜 센터 : 02-469-5426 (휴대폰 : 010-2243-3394)

전자 메일 : sales@opennaru.com