

도커와 쿠버네티스 시작하기

도커와 마이크로서비스 이해

도커 환경 구성 및 설정

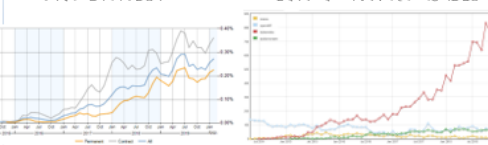
도커 컨테이너 기본 활용

도커와 마이크로서비스 이해

쿠버네티스 이슈

- 컨테이너 기반 오픈 소스 가상화 프로젝트인 쿠버네티스를 지원하는 기업이 증가
- IBM, 액센추어, SAP 등 글로벌 기업이 쿠버네티스 인증 취득
- 삼성SDS 같은 대기업이 정식 퍼드너 자격 서비스 제공

쿠버네티스 엔지니어 구인 증가



컨테이너 오케스트레이션에서 쿠버네티스 사용 비율 급증

<https://www.pitchbook.com/in-depth/kubernetes-2019-market-size-forecast>

-3-

www.boanproject.com

도커 환경 구성 및 설정

우분투 환경 구축

- Ubuntu를 사용
- VirtualBox에서 Ubuntu 설치 후 사용
- 시간 관계상 이미 설치된 파일을 사용
- 설치하는 다음 영상을 참고 (<https://youtu.be/ksqr06lHs>)



-4-

www.boanproject.com

도커 컨테이너 기본 활용

간단한 HTTP 서버 구현하기

- 간단한 애플리케이션을 구현하면서 도커의 사용방법을 하나씩 파악하기.
- 가장 먼저 할 일은 적절한 이미지를 찾는 일

이미지 검색하기

- 도커에서는 search라는 명령어로 내가 원하는 이미지가 이미 레지스트리에 존재하는 지 확인 가능
- `sudo docker search "tomcat"`

NAME	SEARCHED	STARS	OFFICIAL	AUTOMATED
tomcat	Apache Tomcat is an open source implementation of the Java Servlet and JavaSP APIs.	2296	[OK]	[OK]
tomcat8	Apache Tomcat 8.0.0	46	[OK]	[OK]
tomcat8-jdk8	Apache Tomcat 8.0.0 with Java 8	13	[OK]	[OK]
tomcat8-jdk8-alpine	Apache Tomcat 8.0.0 with Java 8 on Alpine Linux	10	[OK]	[OK]
tomcat8-jdk8-debian	Apache Tomcat 8.0.0 with Java 8 on Debian	10	[OK]	[OK]
tomcat8-jdk8-ubuntu	Apache Tomcat 8.0.0 with Java 8 on Ubuntu	10	[OK]	[OK]
tomcat8-jdk8-centos	Apache Tomcat 8.0.0 with Java 8 on CentOS	10	[OK]	[OK]
tomcat8-jdk8-rhel	Apache Tomcat 8.0.0 with Java 8 on RHEL	10	[OK]	[OK]
tomcat8-jdk8-suse	Apache Tomcat 8.0.0 with Java 8 on SUSE	10	[OK]	[OK]
tomcat8-jdk8-fedora	Apache Tomcat 8.0.0 with Java 8 on Fedora	10	[OK]	[OK]
tomcat8-jdk8-rockylinux	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux	10	[OK]	[OK]
tomcat8-jdk8-alma	Apache Tomcat 8.0.0 with Java 8 on AlmaLinux	10	[OK]	[OK]
tomcat8-jdk8-openvulnix	Apache Tomcat 8.0.0 with Java 8 on OpenVulnix	10	[OK]	[OK]
tomcat8-jdk8-oraclelinux	Apache Tomcat 8.0.0 with Java 8 on Oracle Linux	10	[OK]	[OK]
tomcat8-jdk8-rockylinux8	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 8	10	[OK]	[OK]
tomcat8-jdk8-rockylinux9	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 9	10	[OK]	[OK]
tomcat8-jdk8-rockylinux10	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 10	10	[OK]	[OK]
tomcat8-jdk8-rockylinux11	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 11	10	[OK]	[OK]
tomcat8-jdk8-rockylinux12	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 12	10	[OK]	[OK]
tomcat8-jdk8-rockylinux13	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 13	10	[OK]	[OK]
tomcat8-jdk8-rockylinux14	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 14	10	[OK]	[OK]
tomcat8-jdk8-rockylinux15	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 15	10	[OK]	[OK]
tomcat8-jdk8-rockylinux16	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 16	10	[OK]	[OK]
tomcat8-jdk8-rockylinux17	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 17	10	[OK]	[OK]
tomcat8-jdk8-rockylinux18	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 18	10	[OK]	[OK]
tomcat8-jdk8-rockylinux19	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 19	10	[OK]	[OK]
tomcat8-jdk8-rockylinux20	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 20	10	[OK]	[OK]
tomcat8-jdk8-rockylinux21	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 21	10	[OK]	[OK]
tomcat8-jdk8-rockylinux22	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 22	10	[OK]	[OK]
tomcat8-jdk8-rockylinux23	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 23	10	[OK]	[OK]
tomcat8-jdk8-rockylinux24	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 24	10	[OK]	[OK]
tomcat8-jdk8-rockylinux25	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 25	10	[OK]	[OK]
tomcat8-jdk8-rockylinux26	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 26	10	[OK]	[OK]
tomcat8-jdk8-rockylinux27	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 27	10	[OK]	[OK]
tomcat8-jdk8-rockylinux28	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 28	10	[OK]	[OK]
tomcat8-jdk8-rockylinux29	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 29	10	[OK]	[OK]
tomcat8-jdk8-rockylinux30	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 30	10	[OK]	[OK]
tomcat8-jdk8-rockylinux31	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 31	10	[OK]	[OK]
tomcat8-jdk8-rockylinux32	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 32	10	[OK]	[OK]
tomcat8-jdk8-rockylinux33	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 33	10	[OK]	[OK]
tomcat8-jdk8-rockylinux34	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 34	10	[OK]	[OK]
tomcat8-jdk8-rockylinux35	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 35	10	[OK]	[OK]
tomcat8-jdk8-rockylinux36	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 36	10	[OK]	[OK]
tomcat8-jdk8-rockylinux37	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 37	10	[OK]	[OK]
tomcat8-jdk8-rockylinux38	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 38	10	[OK]	[OK]
tomcat8-jdk8-rockylinux39	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 39	10	[OK]	[OK]
tomcat8-jdk8-rockylinux40	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 40	10	[OK]	[OK]
tomcat8-jdk8-rockylinux41	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 41	10	[OK]	[OK]
tomcat8-jdk8-rockylinux42	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 42	10	[OK]	[OK]
tomcat8-jdk8-rockylinux43	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 43	10	[OK]	[OK]
tomcat8-jdk8-rockylinux44	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 44	10	[OK]	[OK]
tomcat8-jdk8-rockylinux45	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 45	10	[OK]	[OK]
tomcat8-jdk8-rockylinux46	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 46	10	[OK]	[OK]
tomcat8-jdk8-rockylinux47	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 47	10	[OK]	[OK]
tomcat8-jdk8-rockylinux48	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 48	10	[OK]	[OK]
tomcat8-jdk8-rockylinux49	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 49	10	[OK]	[OK]
tomcat8-jdk8-rockylinux50	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 50	10	[OK]	[OK]
tomcat8-jdk8-rockylinux51	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 51	10	[OK]	[OK]
tomcat8-jdk8-rockylinux52	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 52	10	[OK]	[OK]
tomcat8-jdk8-rockylinux53	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 53	10	[OK]	[OK]
tomcat8-jdk8-rockylinux54	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 54	10	[OK]	[OK]
tomcat8-jdk8-rockylinux55	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 55	10	[OK]	[OK]
tomcat8-jdk8-rockylinux56	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 56	10	[OK]	[OK]
tomcat8-jdk8-rockylinux57	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 57	10	[OK]	[OK]
tomcat8-jdk8-rockylinux58	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 58	10	[OK]	[OK]
tomcat8-jdk8-rockylinux59	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 59	10	[OK]	[OK]
tomcat8-jdk8-rockylinux60	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 60	10	[OK]	[OK]
tomcat8-jdk8-rockylinux61	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 61	10	[OK]	[OK]
tomcat8-jdk8-rockylinux62	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 62	10	[OK]	[OK]
tomcat8-jdk8-rockylinux63	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 63	10	[OK]	[OK]
tomcat8-jdk8-rockylinux64	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 64	10	[OK]	[OK]
tomcat8-jdk8-rockylinux65	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 65	10	[OK]	[OK]
tomcat8-jdk8-rockylinux66	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 66	10	[OK]	[OK]
tomcat8-jdk8-rockylinux67	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 67	10	[OK]	[OK]
tomcat8-jdk8-rockylinux68	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 68	10	[OK]	[OK]
tomcat8-jdk8-rockylinux69	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 69	10	[OK]	[OK]
tomcat8-jdk8-rockylinux70	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 70	10	[OK]	[OK]
tomcat8-jdk8-rockylinux71	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 71	10	[OK]	[OK]
tomcat8-jdk8-rockylinux72	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 72	10	[OK]	[OK]
tomcat8-jdk8-rockylinux73	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 73	10	[OK]	[OK]
tomcat8-jdk8-rockylinux74	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 74	10	[OK]	[OK]
tomcat8-jdk8-rockylinux75	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 75	10	[OK]	[OK]
tomcat8-jdk8-rockylinux76	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 76	10	[OK]	[OK]
tomcat8-jdk8-rockylinux77	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 77	10	[OK]	[OK]
tomcat8-jdk8-rockylinux78	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 78	10	[OK]	[OK]
tomcat8-jdk8-rockylinux79	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 79	10	[OK]	[OK]
tomcat8-jdk8-rockylinux80	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 80	10	[OK]	[OK]
tomcat8-jdk8-rockylinux81	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 81	10	[OK]	[OK]
tomcat8-jdk8-rockylinux82	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 82	10	[OK]	[OK]
tomcat8-jdk8-rockylinux83	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 83	10	[OK]	[OK]
tomcat8-jdk8-rockylinux84	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 84	10	[OK]	[OK]
tomcat8-jdk8-rockylinux85	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 85	10	[OK]	[OK]
tomcat8-jdk8-rockylinux86	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 86	10	[OK]	[OK]
tomcat8-jdk8-rockylinux87	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 87	10	[OK]	[OK]
tomcat8-jdk8-rockylinux88	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 88	10	[OK]	[OK]
tomcat8-jdk8-rockylinux89	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 89	10	[OK]	[OK]
tomcat8-jdk8-rockylinux90	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 90	10	[OK]	[OK]
tomcat8-jdk8-rockylinux91	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 91	10	[OK]	[OK]
tomcat8-jdk8-rockylinux92	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 92	10	[OK]	[OK]
tomcat8-jdk8-rockylinux93	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 93	10	[OK]	[OK]
tomcat8-jdk8-rockylinux94	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 94	10	[OK]	[OK]
tomcat8-jdk8-rockylinux95	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 95	10	[OK]	[OK]
tomcat8-jdk8-rockylinux96	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 96	10	[OK]	[OK]
tomcat8-jdk8-rockylinux97	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 97	10	[OK]	[OK]
tomcat8-jdk8-rockylinux98	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 98	10	[OK]	[OK]
tomcat8-jdk8-rockylinux99	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 99	10	[OK]	[OK]
tomcat8-jdk8-rockylinux100	Apache Tomcat 8.0.0 with Java 8 on Rocky Linux 100	10	[OK]	[OK]

-5-

www.boanproject.com

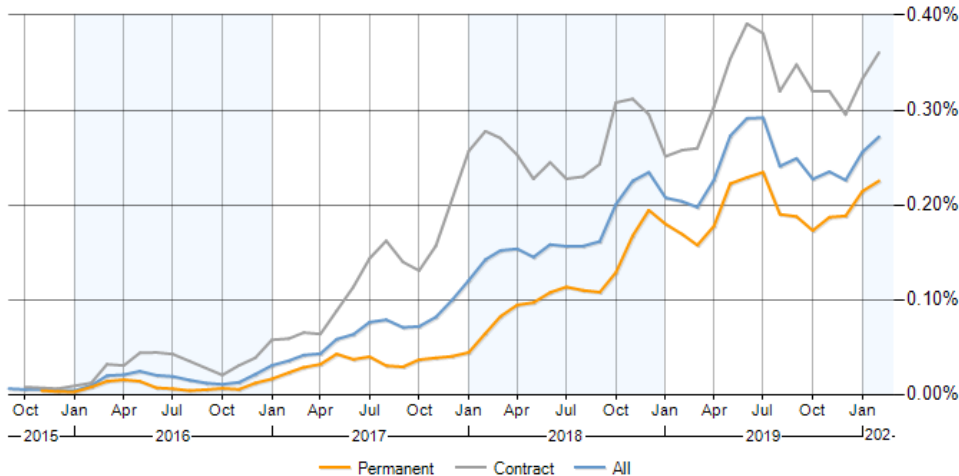
도커와 마이크로서비스 이해

도커와 마이크로서비스 이해

쿠버네티스 이슈

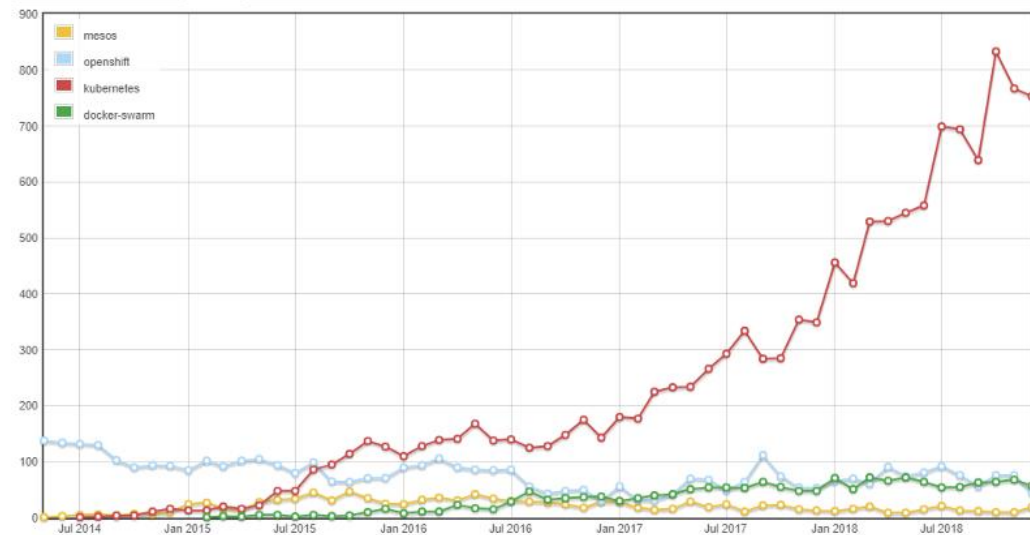
- 컨테이너 기반 오픈 소스 가상화 프로젝트인 쿠버네티스를 지원하는 기업이 증가
- IBM, 액센추어, SAP 등 글로벌 기업이 쿠버네티스 인증 취득
- 삼성SDS 같은 대기업이 정식 파트너 자격 서비스 제공

쿠버네티스 엔지니어 구인 증가



<https://www.itjobswatch.co.uk/jobs/uk/kubernetes%20engineer.do>

컨테이너 오케스트라에서 쿠버네티스 사용 비율 급증

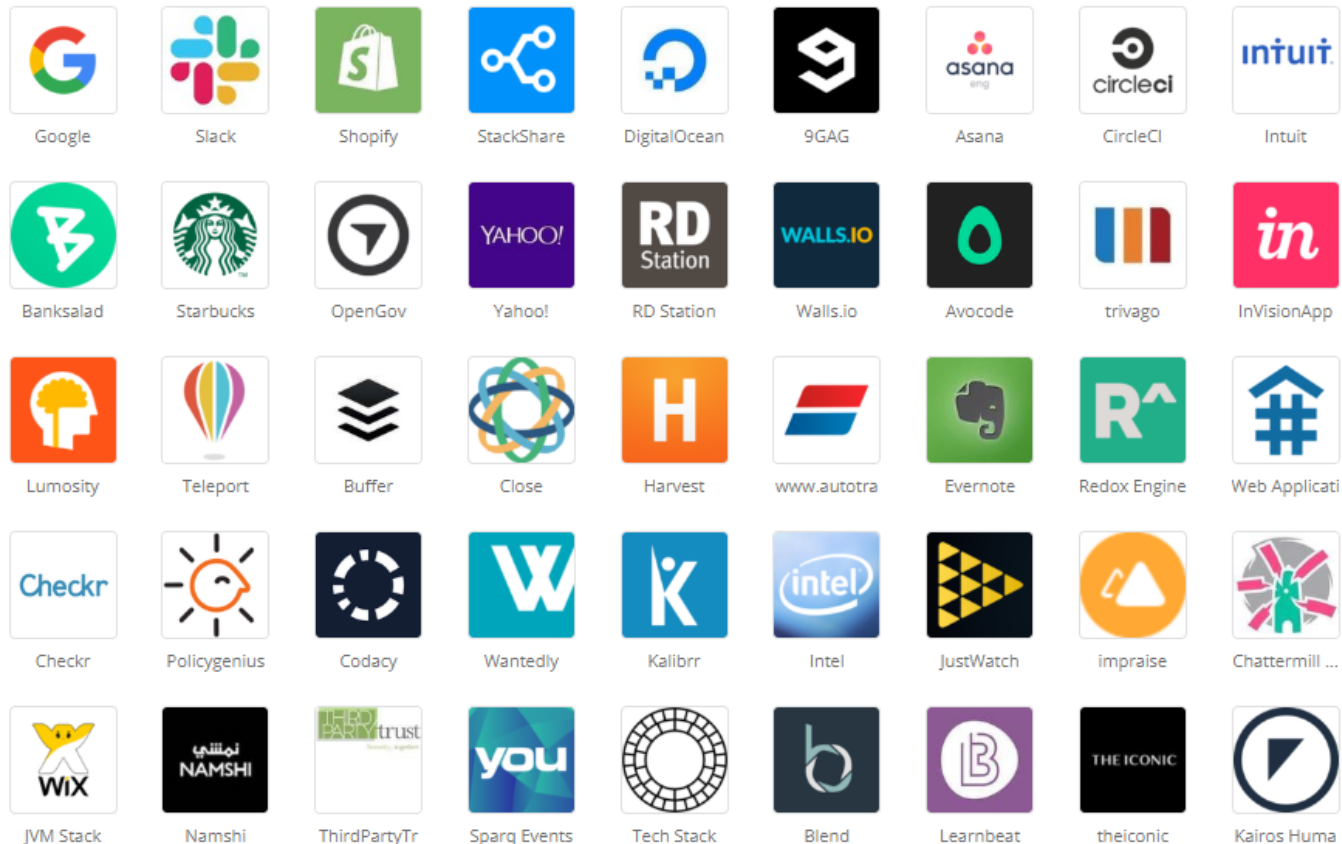


<https://medium.com/@rdodev/saved-you-an-analyst-read-on-kubernetes-growth-2019-edition-d34a3e5a8755>

도커와 마이크로서비스 이해

▶ 쿠버네티스를 도입한 회사들

- 1711 개 회사는 구글, 슬랙, Shopify를 포함한 기술 스택에서 Kubernetes를 사용가상화 점유율 9.1% 이상



<https://stackshare.io/kubernetes/in-stacks>

도커와 마이크로서비스 이해

▶ 쿠버네티스에서 가장 많이 돌리는 TOP12 애플리케이션 컴포넌트



JVM



redis



etcd

NGINX



PostgreSQL



fluentd



THE
APACHE
SOFTWARE FOUNDATION



elastic



mongoDB®

php fpm



RabbitMQ

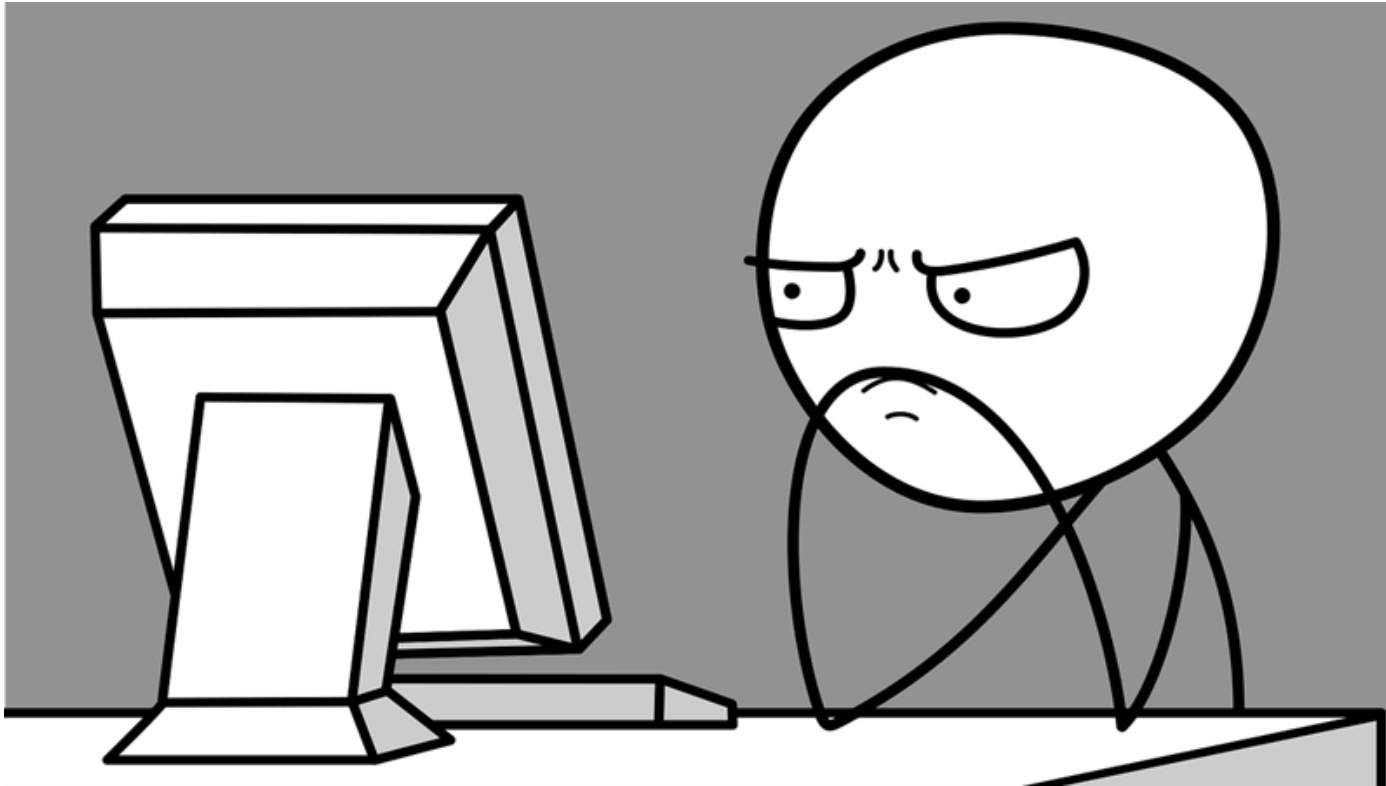


Apache
Tomcat

<https://sysdig.com/blog/2018-docker-usage-report/>

도커와 마이크로서비스 이해

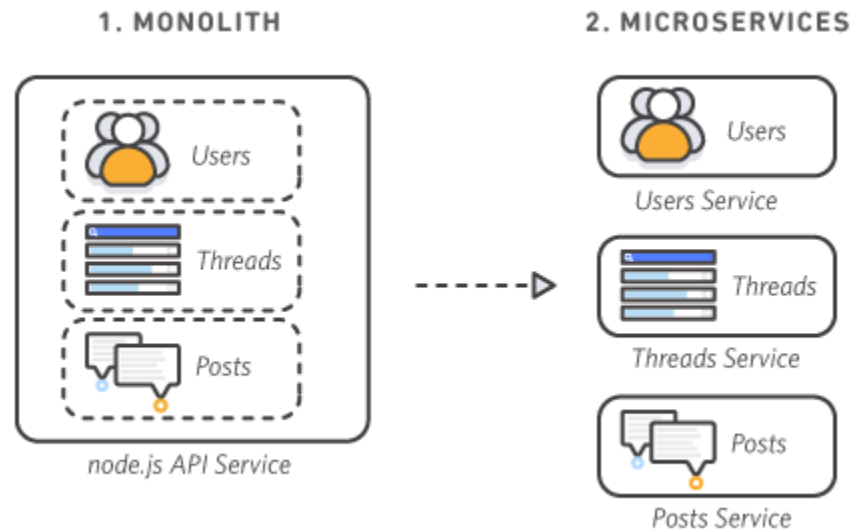
왜 쿠버네티스인가!?



도커와 마이크로서비스 이해

모놀리식(Monolithic architecture) 아키텍처

- 전통적인 아키텍처, 기존에 사용하던 서비스 방법
- 서비스가 하나의 애플리케이션으로 돌아가는 구조
- 기존의 개발 방식을 사용해 개발하여 간단히 배포
- 하나의 서비스 또는 어플리케이션이 하나의 거대한 아키텍처
- 다양한 기능을 동작하는 서비스를 서버에서 실행하여 서비스

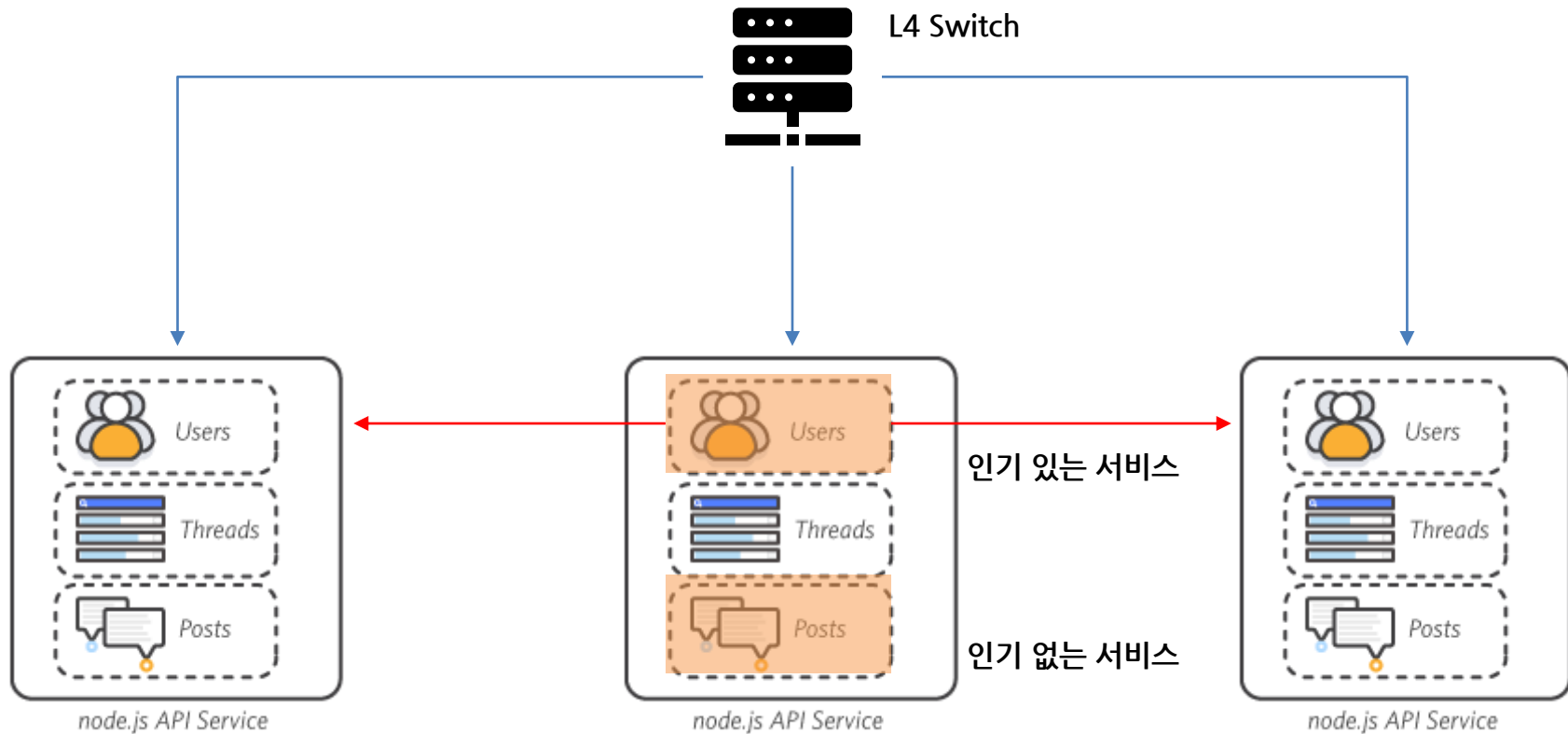


모놀리식 애플리케이션을 마이크로서비스로 분할
<https://aws.amazon.com/ko/microservices/>

도커와 마이크로서비스 이해

▶▶ 단점1. 모놀리식 서비스 아키텍처를 스케일링하면!

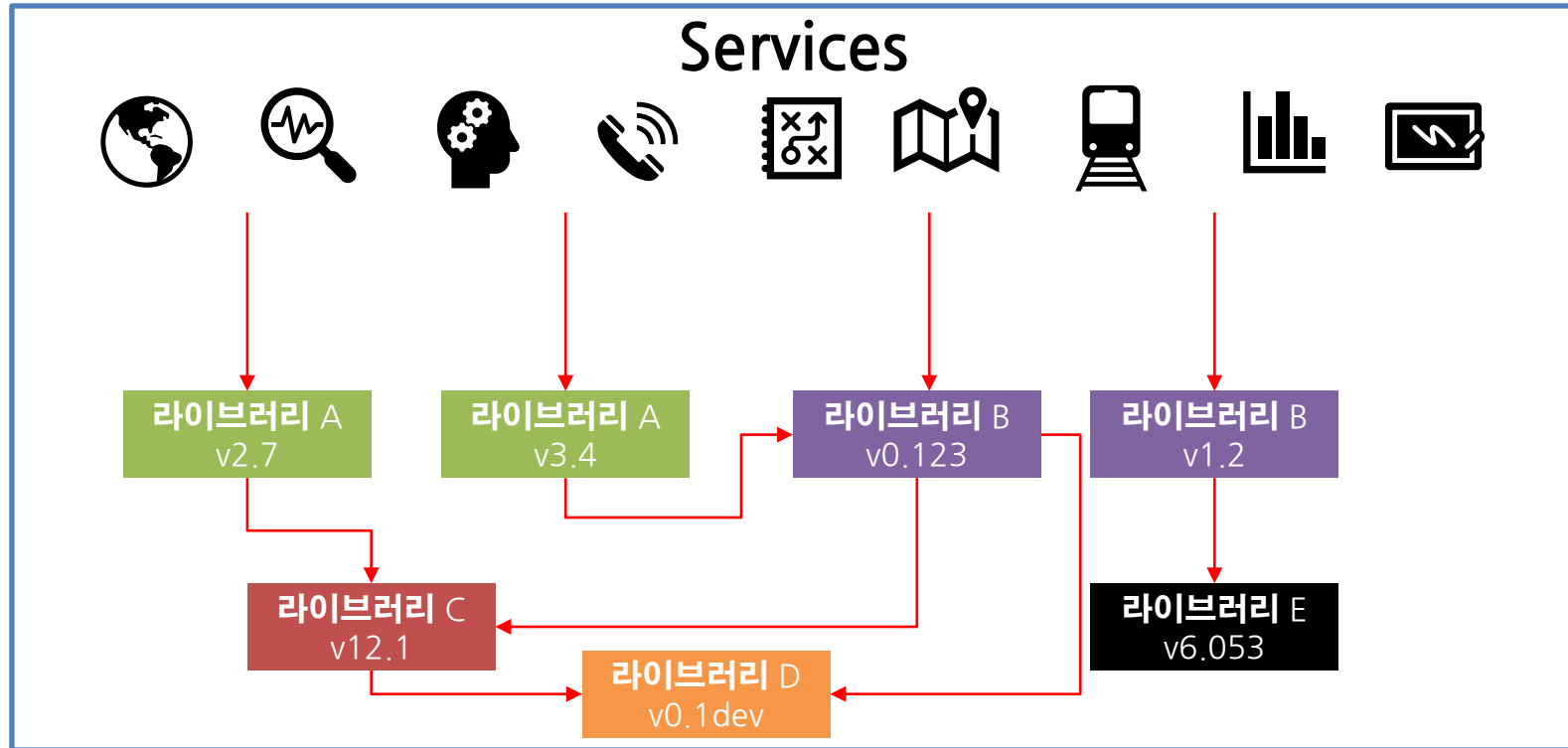
- 기존의 애플리케이션을 그대로 복제하여 로드밸런싱
- 불필요한 서비스까지 모두 복제



도커와 마이크로서비스 이해

▶▶ 단점2. 종속적인 라이브러리의 충돌

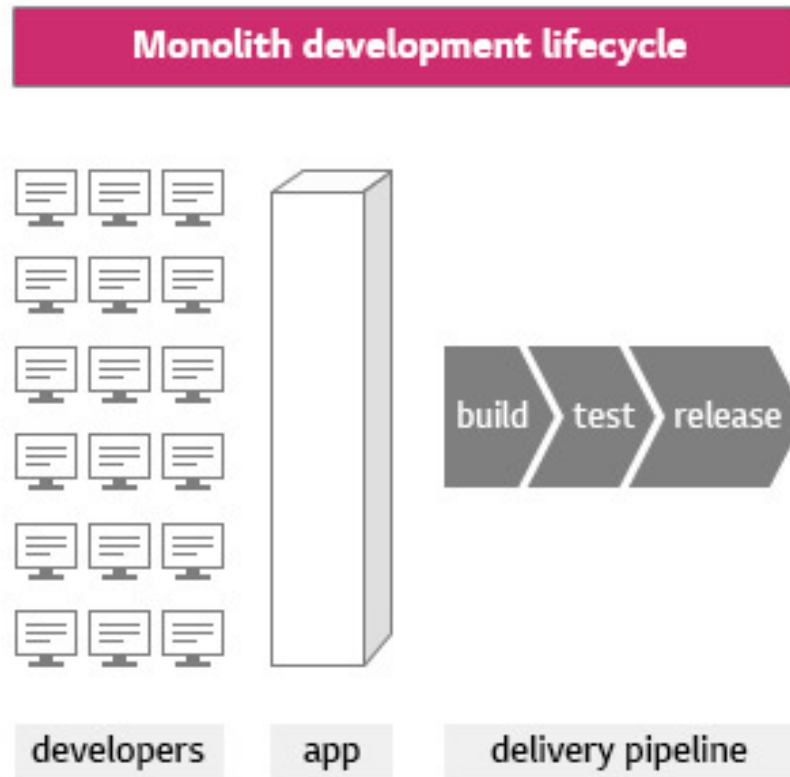
- 각각의 기능들은 서로 다른 기능을 제공하여 버전의 종속성을 필요한 경우가 존재
- 각 기능의 다른 라이브러리를 매업데이트마다 관리하기 매우 어려움



도커와 마이크로서비스 이해

▶ 단점3. 조금만 수정해도 전체 빌드 및 배포 필요

- 소스코드 전체가 하나로써 동작하기 작은 수정만 있더라도 전체를 빌드하여 다시 배포해야 함
- 프로그램의 크기가 어느정도 커지면 한 번만 컴파일해서 전체 테스트를 돌려도 30분 내지 수 시간 소모
- 하루에 버그가 여러 개 순차적으로 발견되면? → 지옥



도커와 마이크로서비스 이해

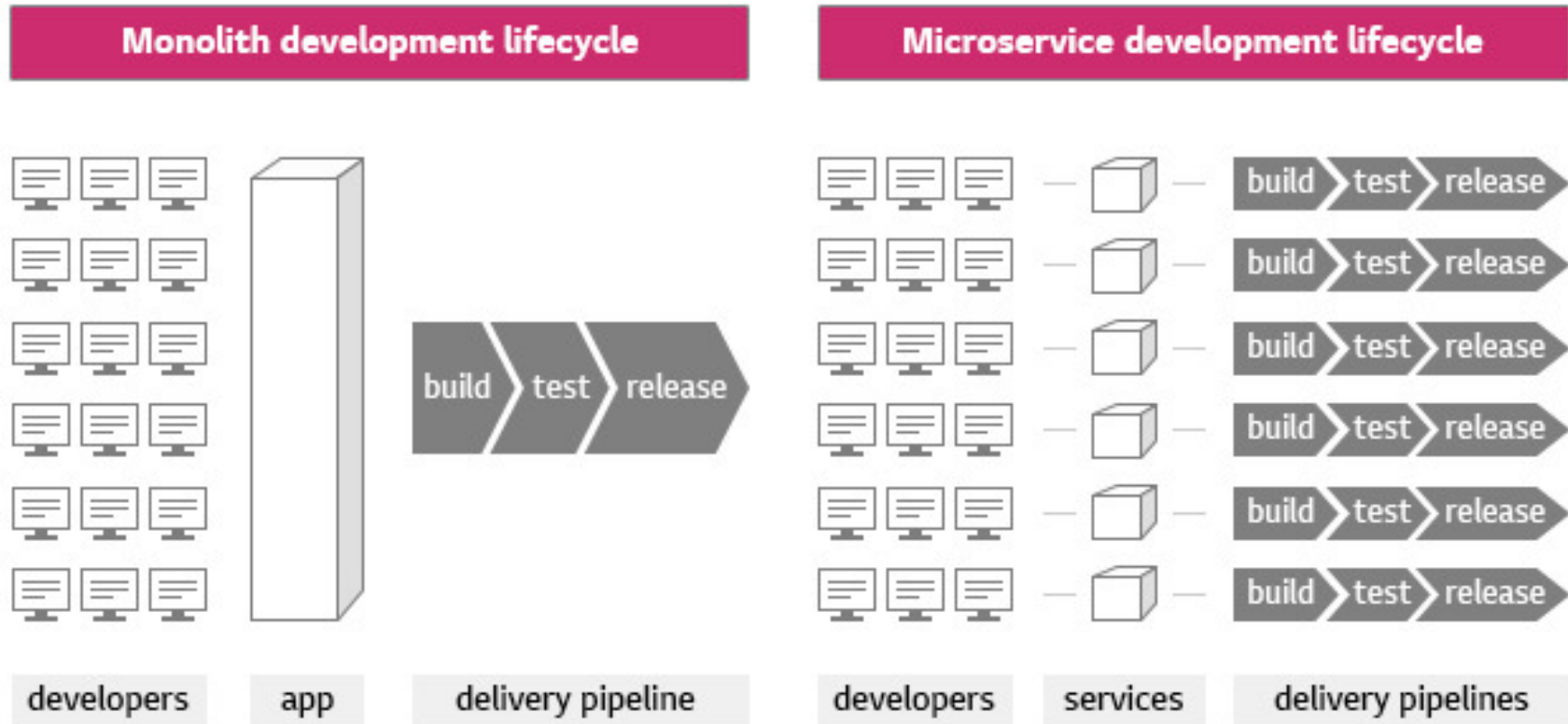
▶ 마이크로서비스 아키텍처

- 모놀리식 아키텍처의 대안으로 반대되는 개념
- 애플리케이션의 각각의 기능을 분리하여 개발 및 관리
- 마이크로서비스 장점
 - 서비스 단위 빠른 개발: 개발자가 특정 비즈니스 로직에 대해서만 집중하여 개발 가능
 - 배포 용이: 개별 서비스 단위로 개발, 패키징, 빌드, 테스트, 배포로 각 서비스마다 유연한 스케줄
 - 서비스 단위 고효율 저비용 Scale-Out 구조: 서비스 단위로 스케일링이 가능하여 불필요한 서비스는 줄이고 더 많은 자원이 필요한 서비스는 확장가능
- 분산 시스템 환경에서 Transaction 보장, 테스트, 배포, 관리 복잡

도커와 마이크로서비스 이해

▶ 마이크로서비스 아키텍처

- 모놀리식 라이프 사이클과 마이크로서비스 라이프 사이클 비교



모놀리식과 마이크로서비스의 개발-빌드-테스트-릴리즈-비교 <https://blog.lgcns.com/1278>

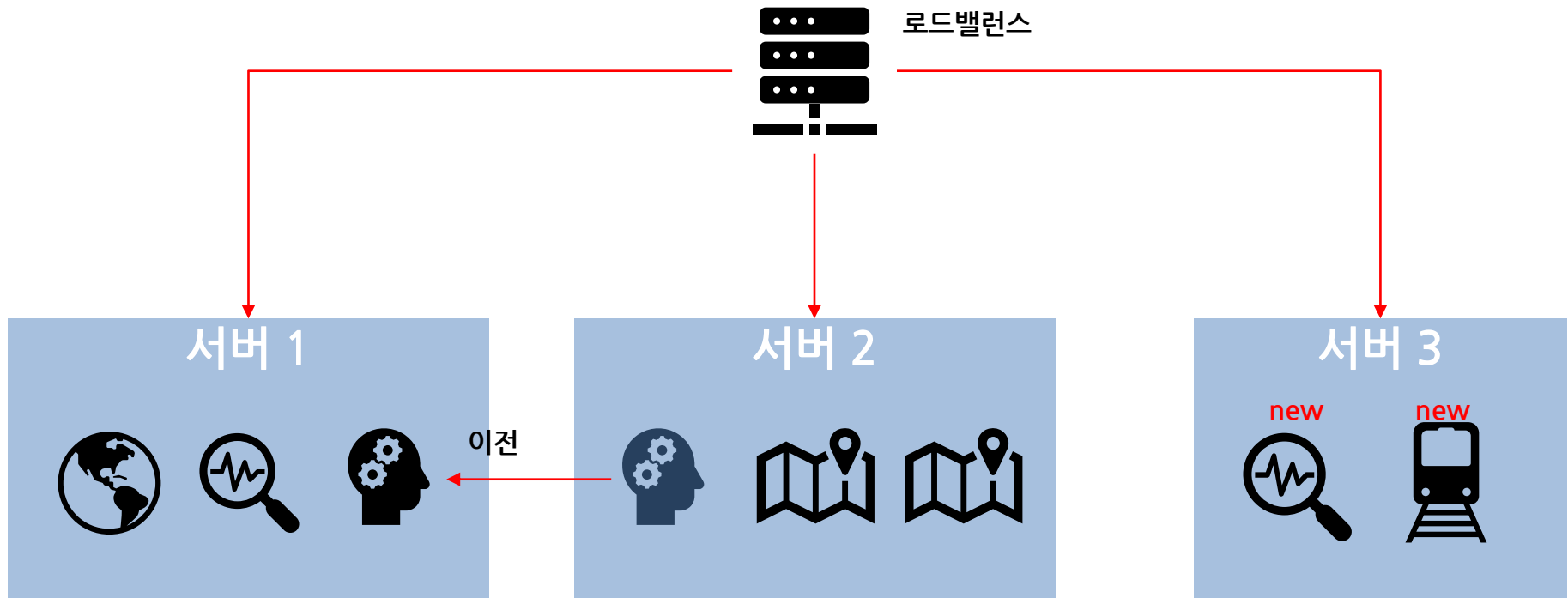
도커와 마이크로서비스 이해

▶ 마이크로서비스 장점

- 서비스 단위 고효율 저비용 Scale-Out 구조

- 서비스 단위로 스케일링이 가능하여 불필요한 서비스는 줄이고 더 많은 자원이 필요한 서비스는 확장가능

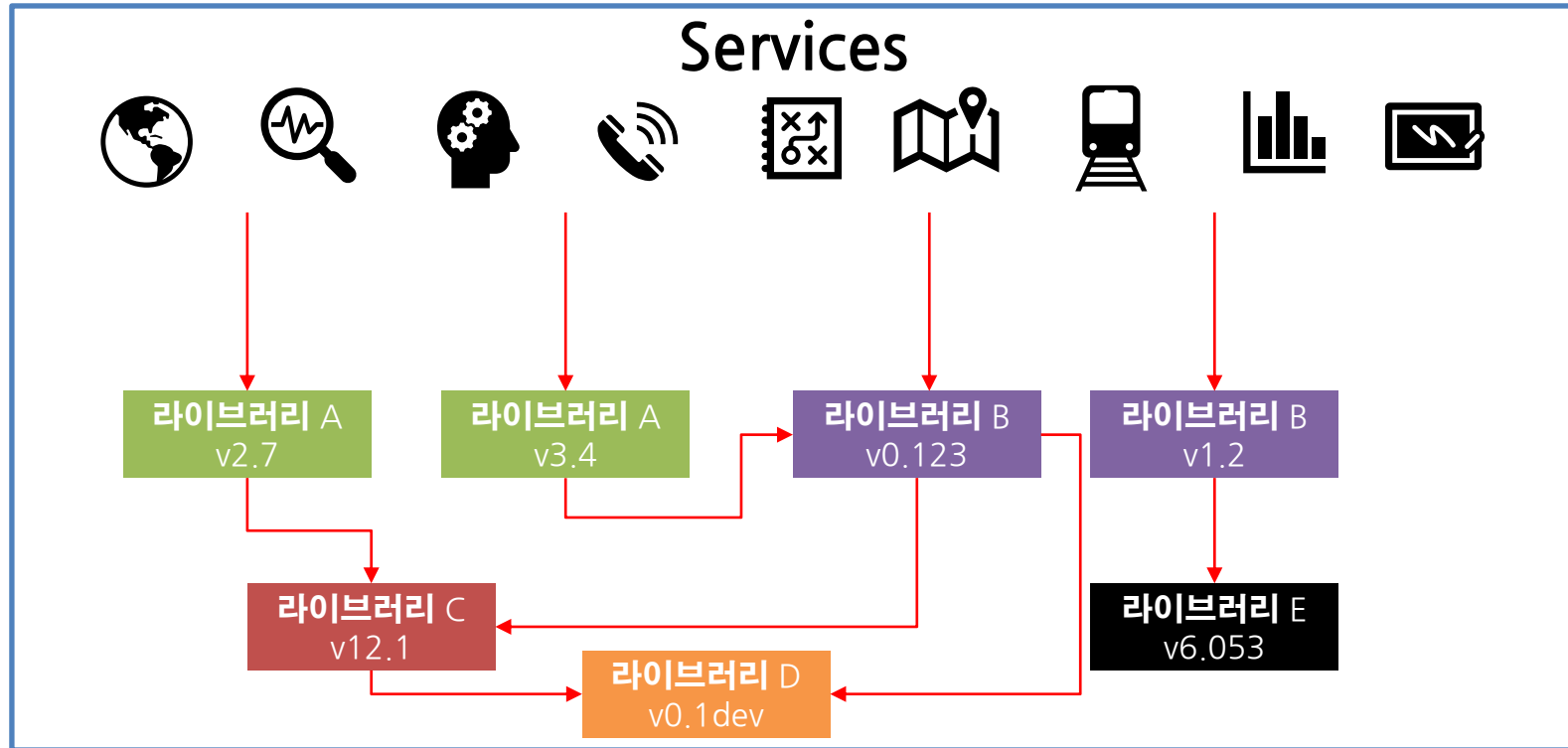
- 라이브러리 종속성 고민 끝!



도커와 마이크로서비스 이해

▶ BUT! 분산 시스템 환경에서 Transaction 보장, 테스트, 배포, 관리 복잡!

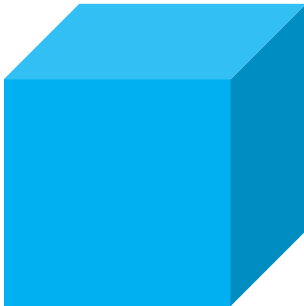
- 이 문제를 해결하면서 마이크로서비스를 할 수 있는 방법이 있을까?



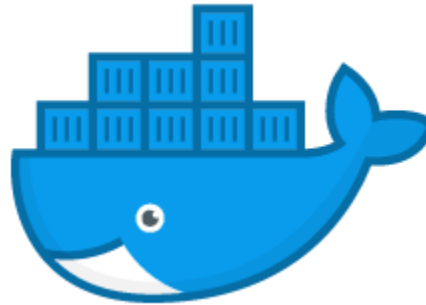
도커와 마이크로서비스 이해

▶ 이미지, 컨테이너, 도커와 쿠버네티스

컨테이너: 용기



도커: 항만 노동자



쿠버네티스: 항해자



도커와 마이크로서비스 이해

컨테이너가 해결한다!

- 컨테이너는 가상머신을 사용해 각 마이크로 서비스를 격리(isolate)하는 기술
- 컨테이너는 가상머신처럼 하드웨어를 전부 구현하지 않기 때문에 매우 빠른 실행 가능
- 프로세스의 문제가 발생할 경우 컨테이너 전체를 조정해야 하기 때문에 컨테이너에 하나의 프로세스를 실행하도록 하는 것이 좋다. (브라우저와 비슷 !)

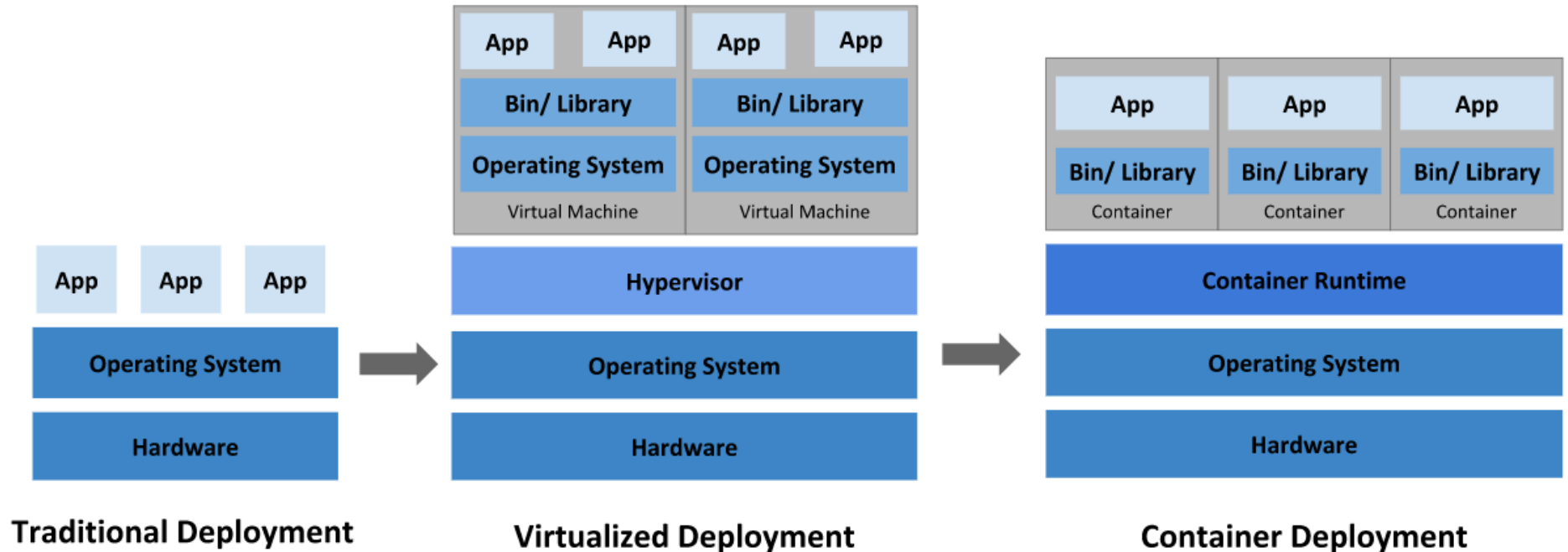


그림 출처: <https://Kubernetes.io/docs/concepts/overview/what-is-Kubernetes/>

도커와 마이크로서비스 이해

VM과 컨테이너의 실행 시 CPU 사용량 비교

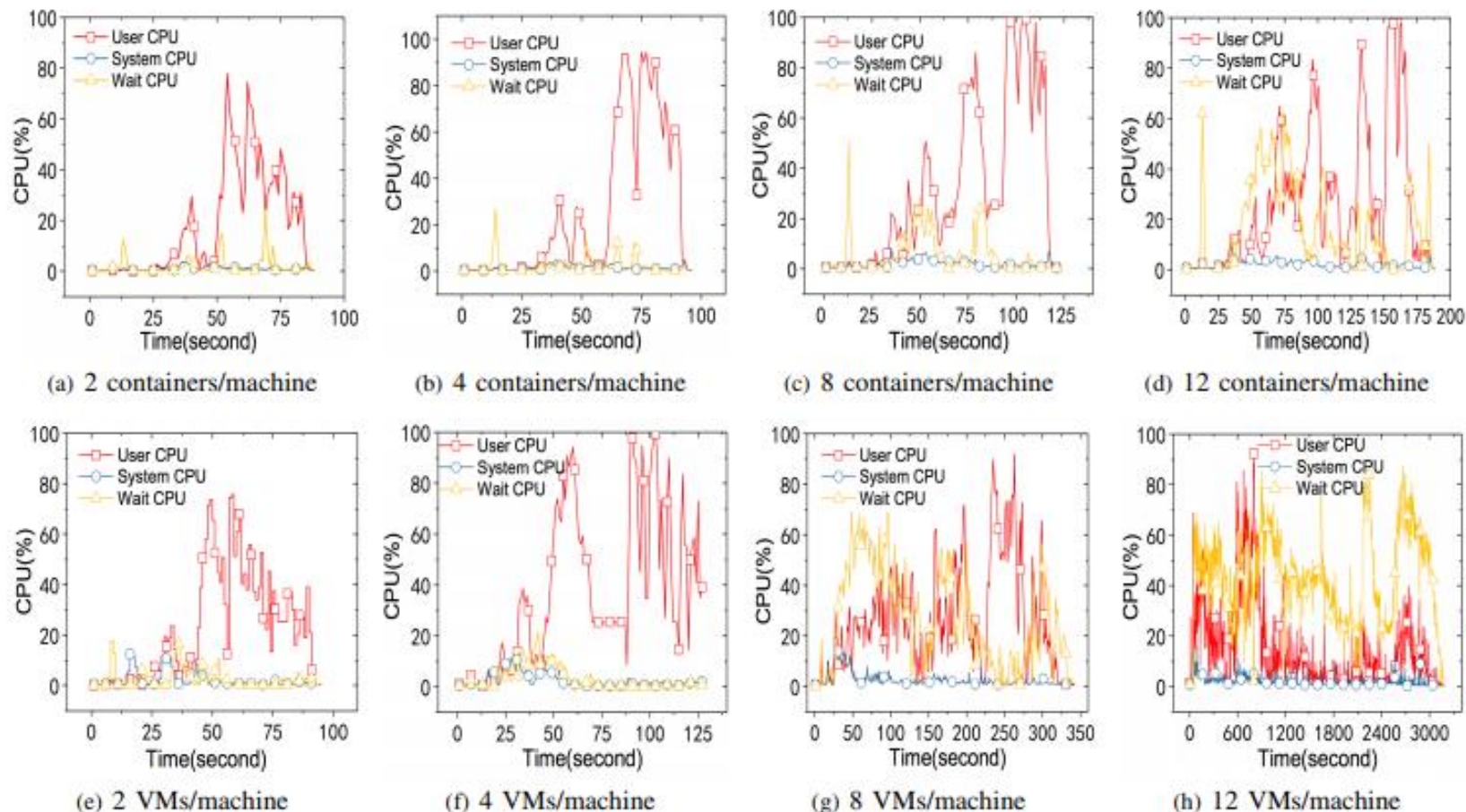


Fig. 6. Average CPU utilization of all the slave machines running with containers and virtual machines, the workload is PageRank.

도커와 마이크로서비스 이해

VM과 컨테이너의 실행 시 CPU 사용량 비교

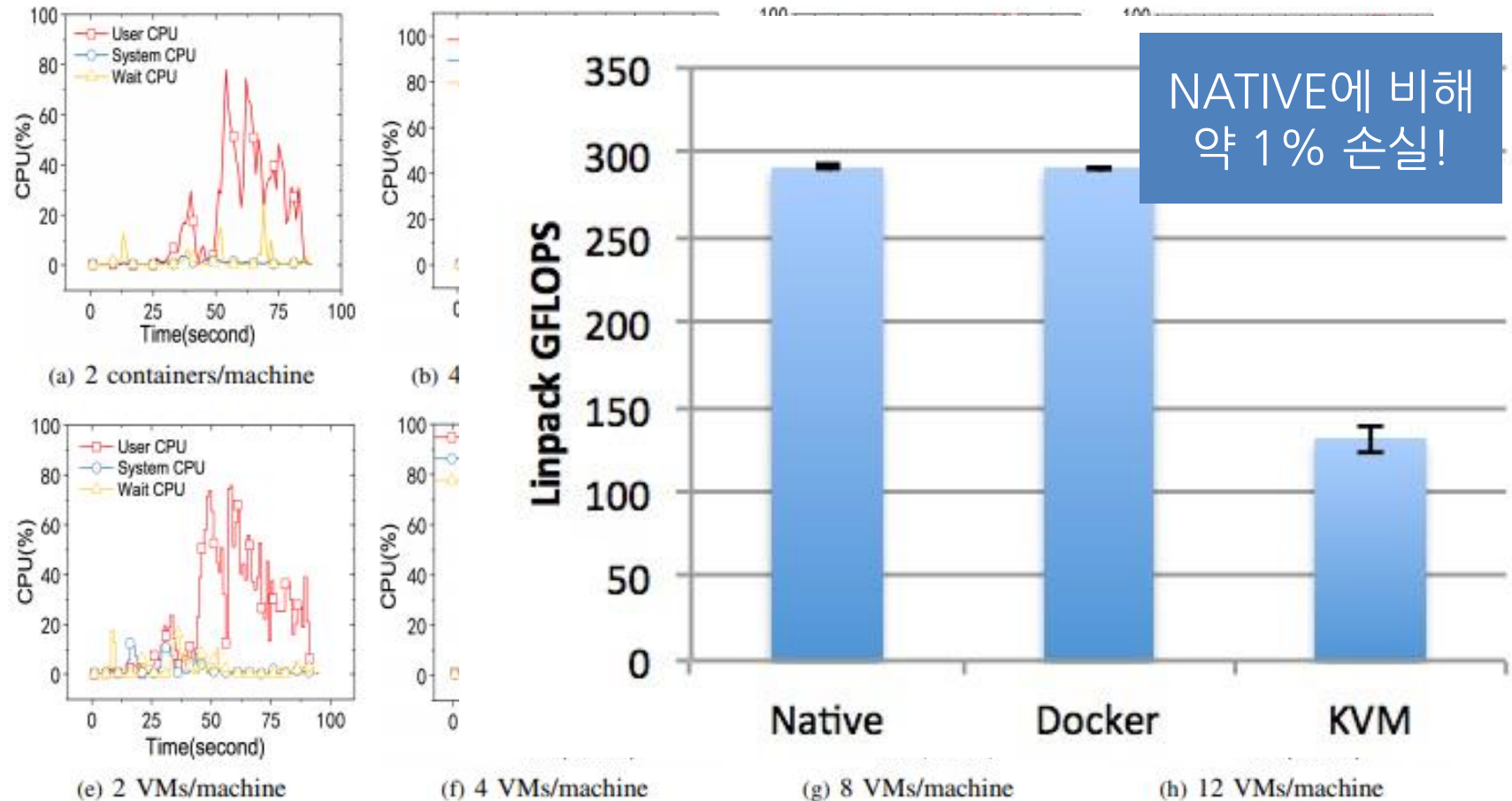


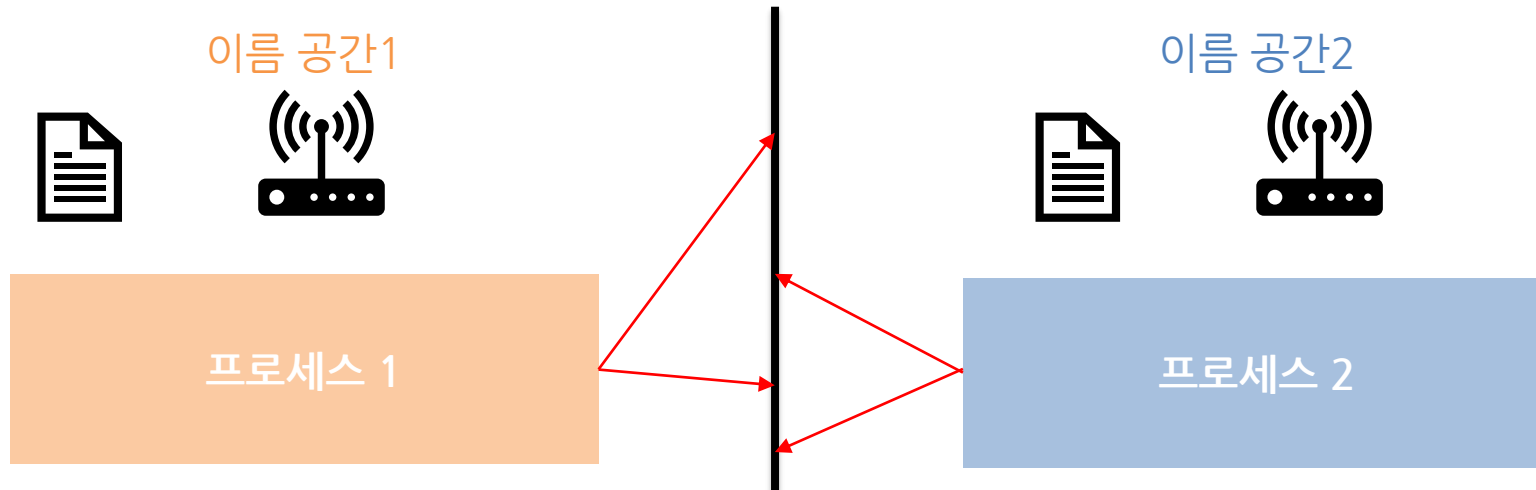
Fig. 6. Average CPU utilization of all the slave machines running with containers and virtual machines, the workload is PageRank.

<https://arxiv.org/pdf/1807.01842.pdf>

도커와 마이크로서비스 이해

컨테이너를 격리하는 기술

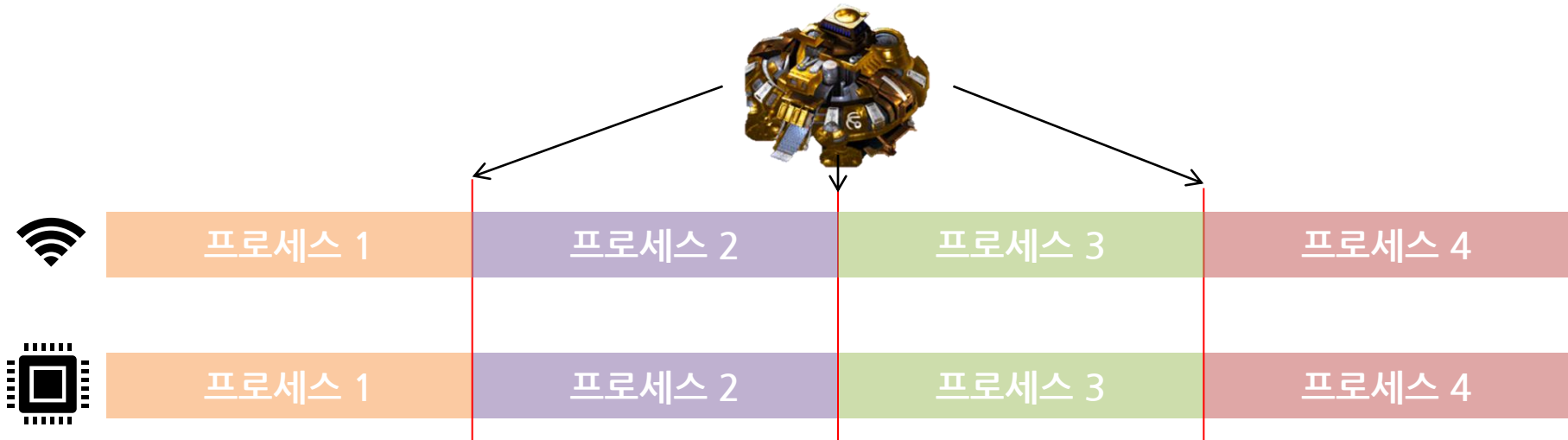
- 리눅스 네임 스페이스: 각 프로세스가 파일 시스템 마운트, 네트워크, 유저(uid), 호스트 네임(uts) 등에 대해 시스템에 독립 뷰를 제공



도커와 마이크로서비스 이해

컨테이너를 격리하는 기술

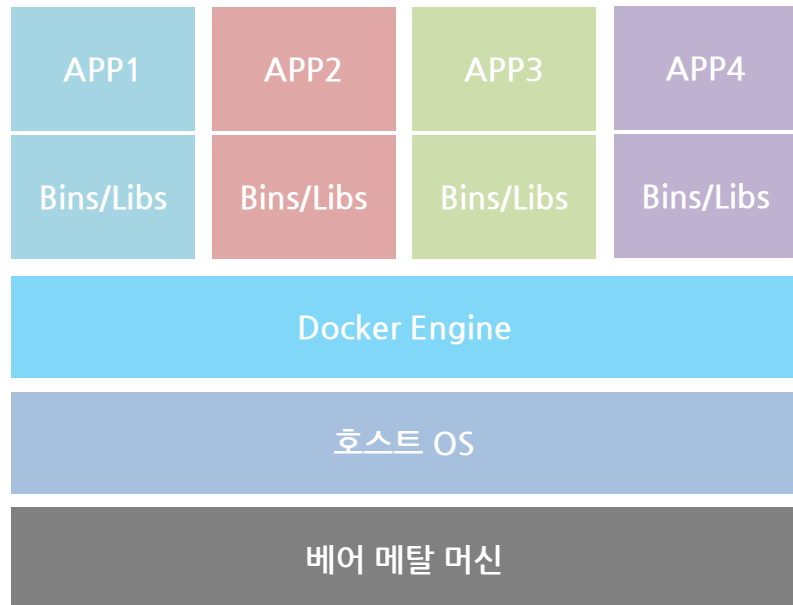
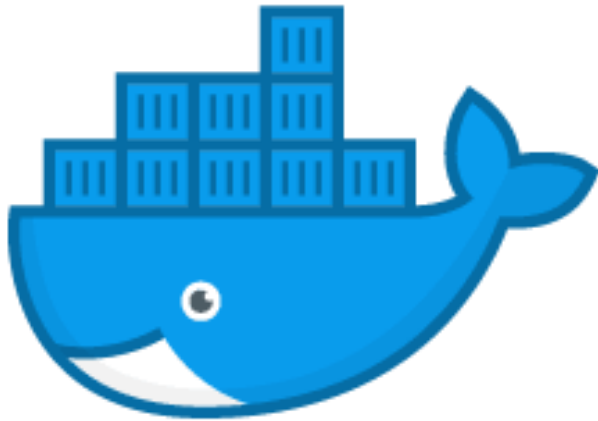
- 리눅스 컨트롤 그룹: 프로세스로 소비할 수 있는 리소스 양(CPU, 메모리, I/O, 네트워크 대역대, device 노드 등)을 제한



도커와 마이크로서비스 이해

도커(Docker)

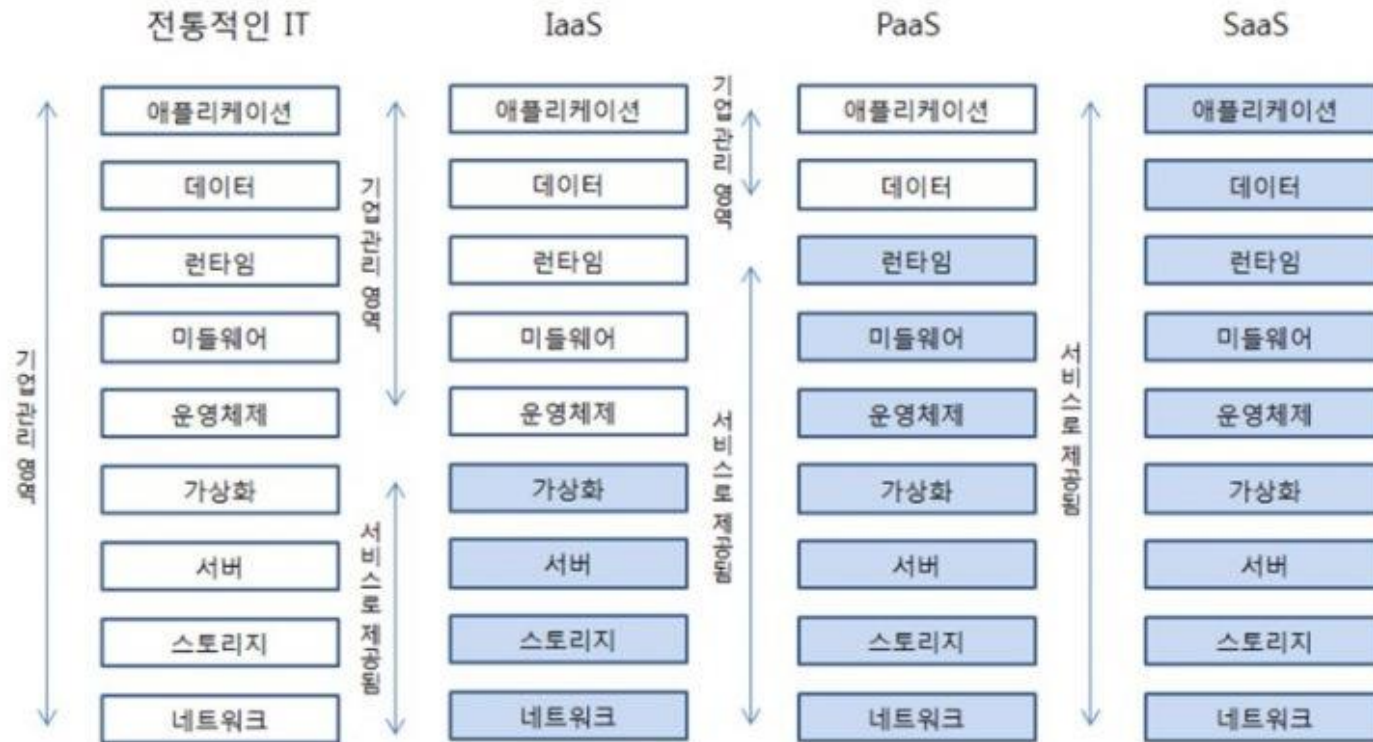
- 컨테이너 기술을 지원하는 다양한 프로젝트 중에 하나
- 컨테이너 기술을 이전에도 있었으나 도커로 인해 알려짐
- 컨테이너 기술의 사실상 표준
- 2014 가장 인기 있는 클라우드 오픈 소스 2위(리눅스 재단 발표)
- 다양한 운영체제에서 사용 가능(리눅스, 윈도우, MacOS)
- 애플리케이션에 국한 되지 않고 의존성 및 파일 시스템까지 패키징하여 빌드, 배포, 실행을 단순화
- 리눅스의 네임 스페이스와 cgroups와 같은 커널 기능을 사용하여 가상화



도커와 마이크로서비스 이해

도커는 다양한 클라우드 서비스 모델과 같이 사용 가능

- 컨테이너: 이미지를 격리하여 독립된 공간에서 실행한 가상 환경
- 이미지: 필요한 프로그램과 라이브러리, 소스를 설치한 뒤 만든 하나의 파일



도커와 마이크로서비스 이해

도커 아키텍처

- Docker engine: 이미지, 네트워크, 디스크 등의 관리 역할
- Containerd: OCI 구현체(주로 runC)를 이용해 container를 관리해주는 daemon
- 두 프로그램이 각각 돌아가기 때문에 Docker Engine을 재시작해도 각 이미지에 영향이 없음

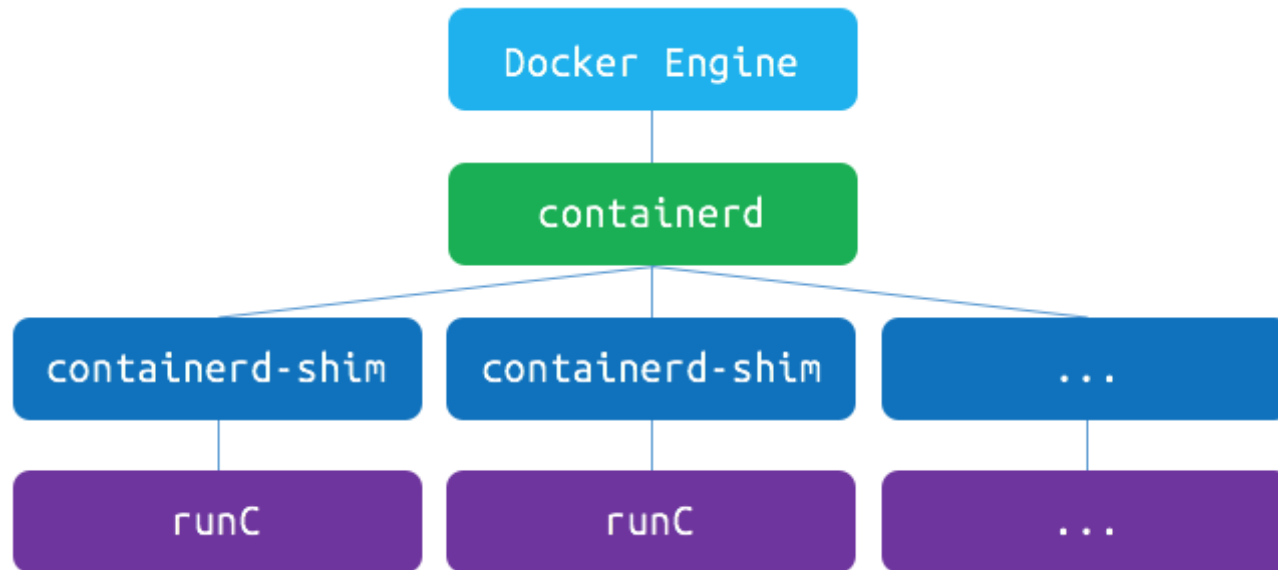
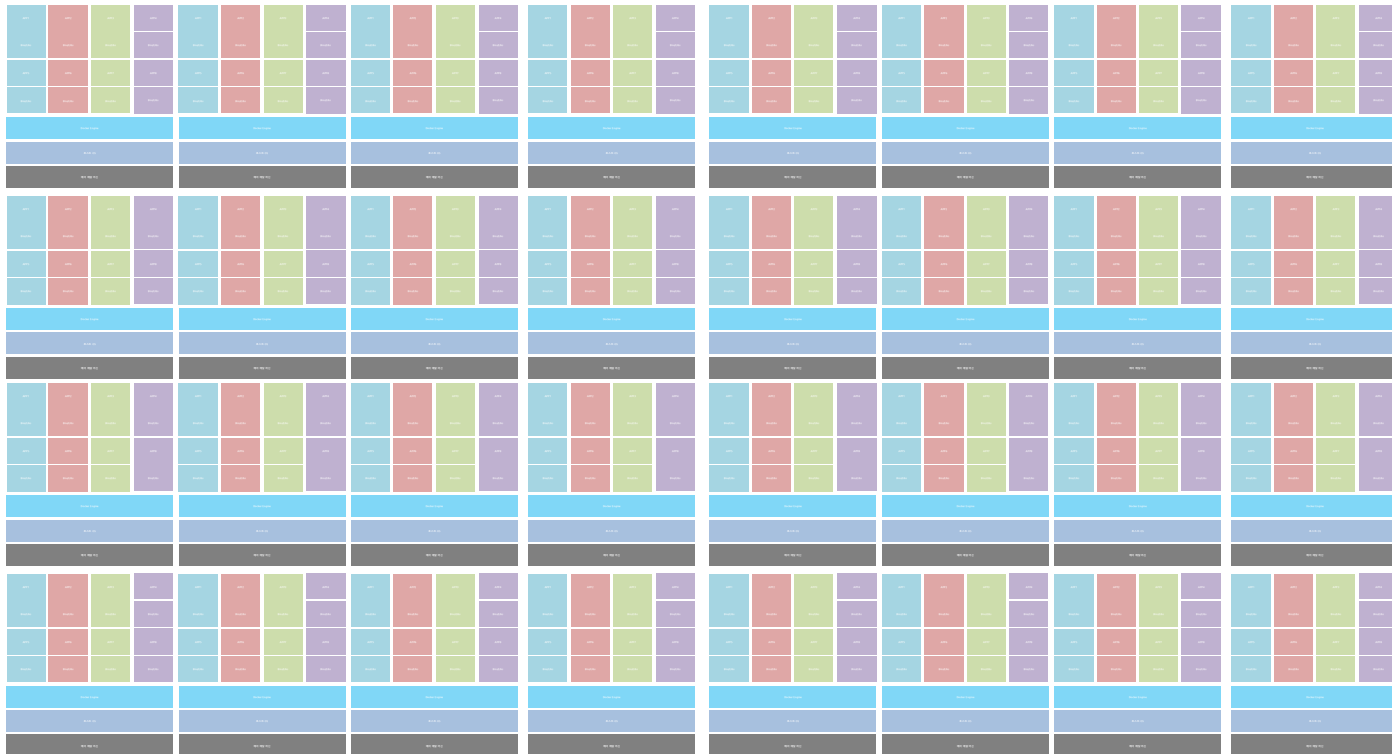


그림 출처: <https://tech.ssut.me/what-even-is-a-container/>

도커와 마이크로서비스 이해

도커의 한계

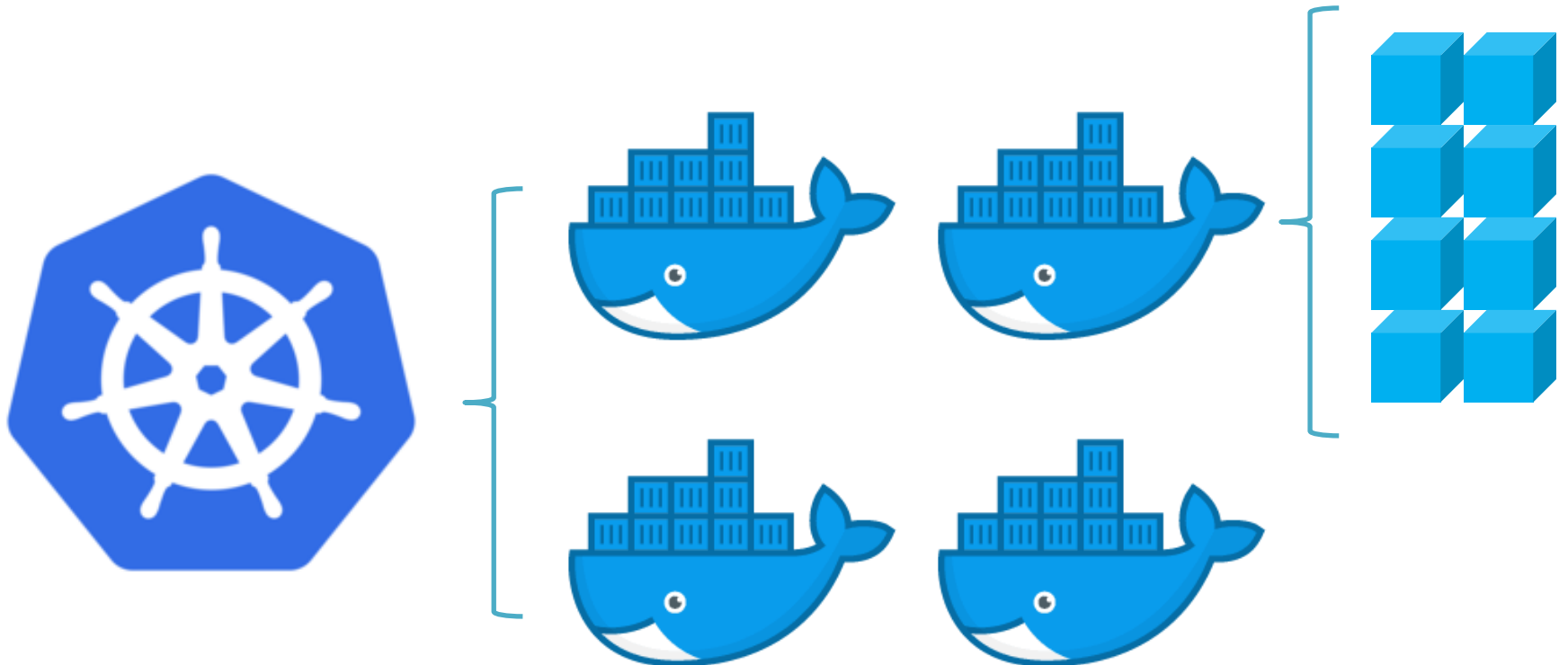
- 서비스가 커지면 커질 수록 관리해야 하는 컨테이너의 양이 급격히 증가
- 도커를 사용하여 관리를 한다 하더라도 쉽지 않은 형태



도커와 마이크로서비스 이해

▶ 쿠버네티스

- 2014년 구글이 오픈 소스 공개
- 구글이 컨테이너 운영 노하우가 담긴 오픈소스
- 고대 그리스어로 항해사라는 의미를 가짐
- 다수의 컨테이너를 자동으로 운영하기 위한 오케스트레이션 도구
- 많은 시스템을 통합, 컨테이너를 다루기 위한 API 제공



도커와 마이크로서비스 이해

▶ 데브옵스(DevOps) 모델

- 데브옵스는 소프트웨어 개발과 IT 운영을 결합한 합성어
- 기존의 분리된 소프트웨어 개발팀과 IT 운영팀의 협업으로 전체 라이프사이클을 함께 관리할 수 있자는 일종의 철학 또는 운동
- 소프트웨어 개발팀과 IT 팀이 더 빠르고 안정적으로 소프트웨어를 빌드, 릴리즈할 수 있도록 두 팀 간의 프로세스를 자동화하는 일련의 과정
- 데브옵스의 이점
 - 속도: 서로의 업무에 대해 더 잘 이해하여 좀더 신속하게 사용자에게 필요한 업데이트를 수행 가능. 개발자는 소비자가 무엇을 원하는지, 운영자는 애플리케이션을 제공하는데 해결할 문제를 인지할 수 있음
 - 신속한 제공: 컨테이너와 마이크로서비스를 사용하면 더 자주 빠르게 릴리즈하는 것이 가능. 더 빠르게 좋은 기능을 제공할 수 있으며 개발자가 운영에 필요한 인프라와 하드웨어에 대해 잘 몰라도 릴리즈가 가능.
 - 개발과 릴리즈가 편해지므로 안정성이 확보, 협업 강화

도커와 마이크로서비스 이해

▶ 개발자와 시스템 관리자의 역할 분담

● 개발자의 관점

- 새로운 기능을 만들어 사용자 경험을 개선하는 것을 좋아함
- 기본 운영체제의 보안 패치나 이와 관련된 모든 것이 최신인지 확인하는 것을 시스템 관리자에게 맡기려 함

● 시스템 관리자의 관점

- 제품 배포와 운영하는 하드웨어 인프라를 담당하며 시스템보안, 활용, 개발자의 우선순위가 높지 않은 측면에 신경을 씀
- 운영 담당자는 모든 애플리케이션 구성 요소의 암묵적 상호 의존성에 대처하기를 원하지 않음
- 기본 운영체제나 인프라를 변경했을 때 애플리케이션 전체 동작에 어떤 영향을 미칠지는 생각치 못함

도커와 마이크로서비스 이해

▶ 노옵스(noops)

- 하드웨어 인프라를 이해하지 않아도 개발자가 직접 애플리케이션을 배포 가능
- 물론 하드웨어 인프라를 관리할 사람은 필요
- 그러나 실행 중인 애플리케이션의 특성을 꼭 알 필요는 없음
- 개발자
 - 애플리케이션을 배포하고 실행하는 단일 플랫폼으로써 실제 하드웨어를 추상화하고 노출하므로 개발자는 시스템 관리자의 도움 없이도 애플리케이션을 구성하고 배포할 수 있다.
- 시스템 관리자
 - 실제로 하드웨어에서 실행되고 있는 애플리케이션에 대해 모르더라도 기본 인프라를 유지하고 가동하는데 집중할 수 있다.

도커와 마이크로서비스 이해

▶ 마이크로서비스 성공 사례

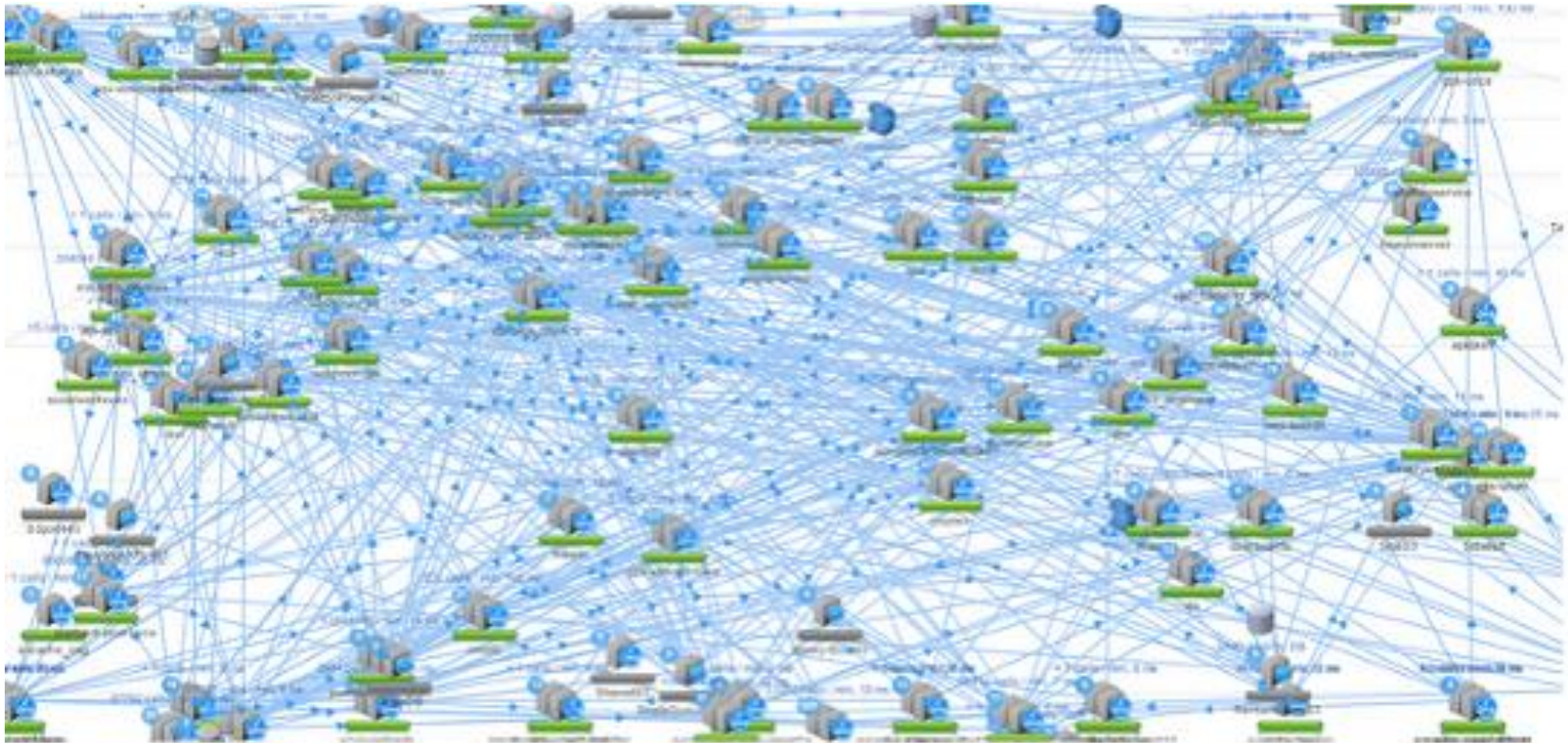
● Netflix 기술 블로그 (2013년)

- 최초의 마이크로서비스 사례
- 수백 개가 넘는 작은 단위의 서비스 운영
- API 기반의 마이크로서비스를 활용한 넷플릭스 아키텍처 의존성 도면



Netflix Technology Blog
Jan 28, 2013 · 5 min read

Follow

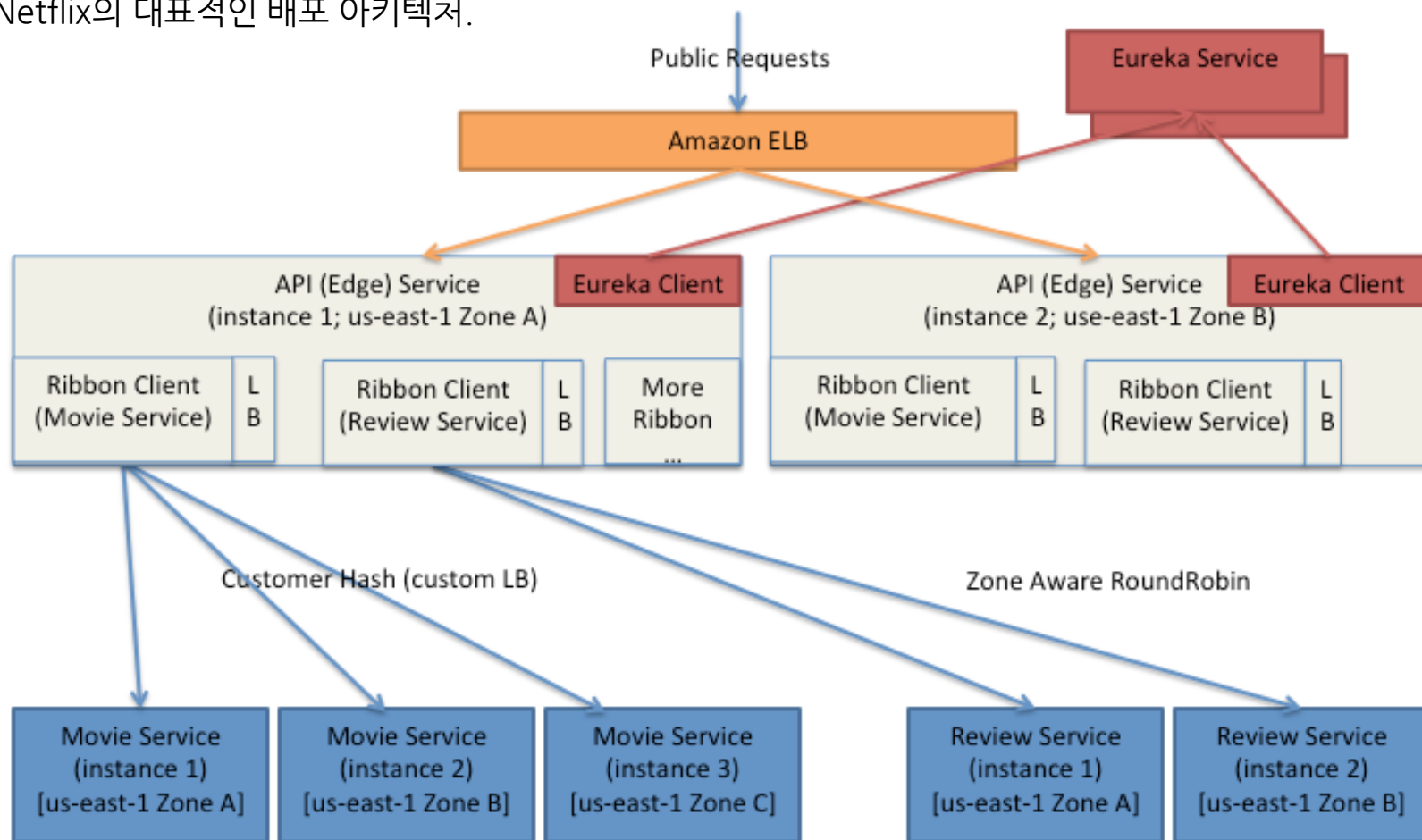


도커와 마이크로서비스 이해

▶ 마이크로서비스 성공 사례

● Service Oriented Architecture

➢ Netflix의 대표적인 배포 아키텍처.

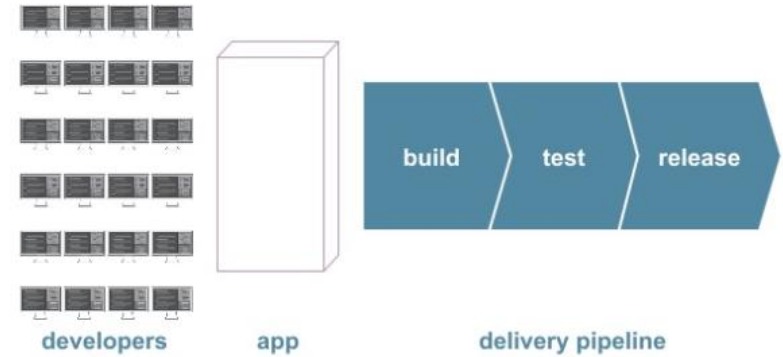
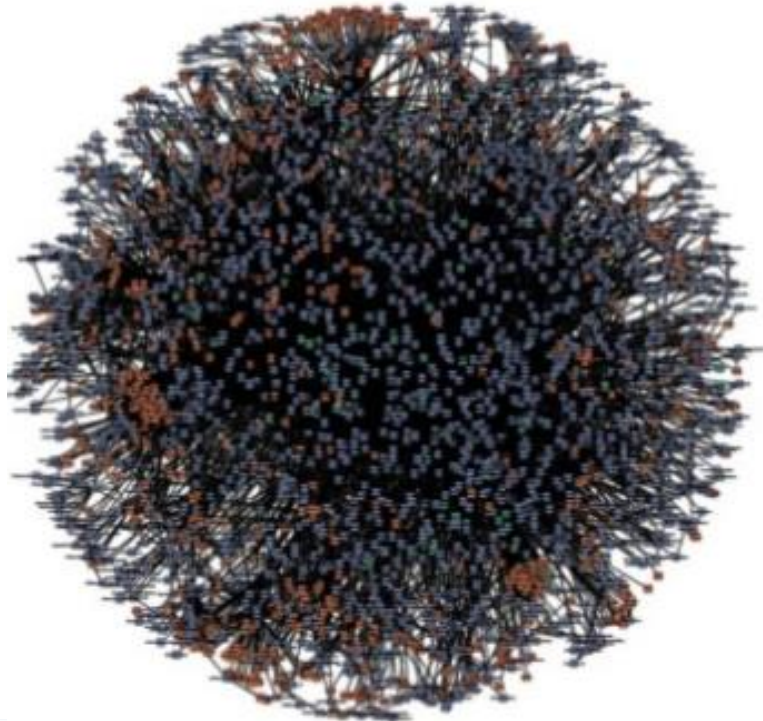


도커와 마이크로서비스 이해

마이크로서비스 성공 사례

● I Love APIs 2015: Microservices at Amazon

- 모놀리식 서비스에서 마이크로서비스로!
- 빌드 타임 감소
- 복잡도 감소
- 수 천개의 DevOps 팀이 자체적으로 배포
- 연간 5천만회(2014년 기준)

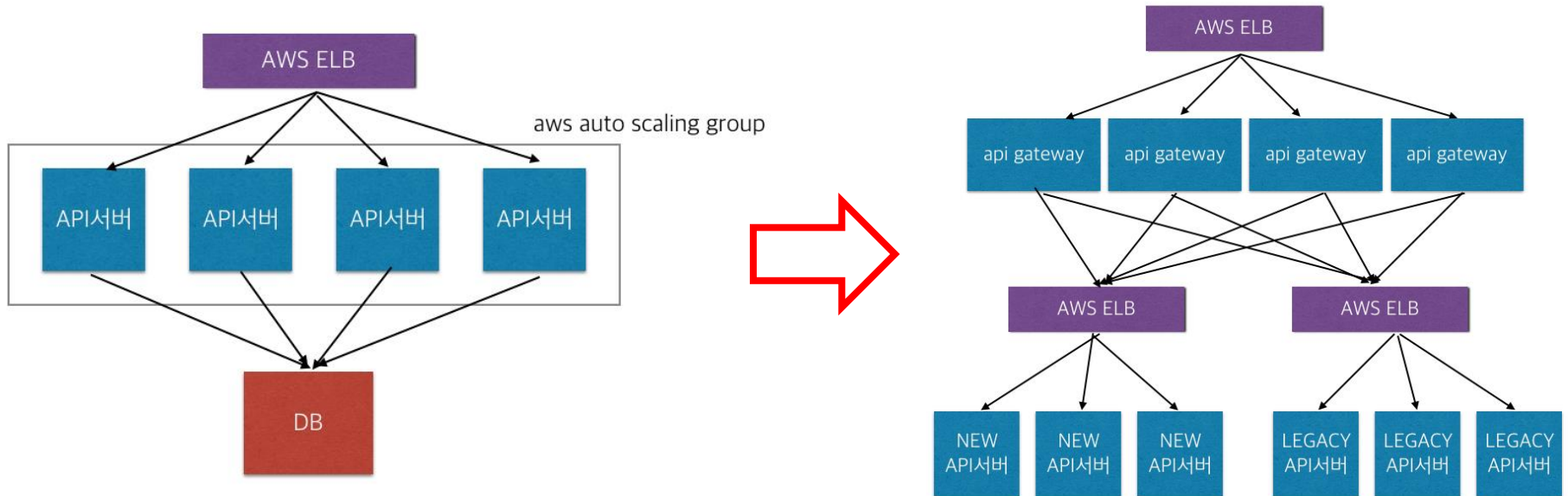


- Single-purpose
- Connect only through APIs
- Connect over HTTPS
- Largely “black boxes” to each other
- “Microservices”

도커와 마이크로서비스 이해

마이크로서비스 성공 사례

- 우아한 형제들(<http://woowabros.github.io/r&d/2017/06/13/apigateway.html>)
 - 모놀리식 아키텍처에서 마이크로서비스 전환기를 거쳤던 과도기의 구조



도커와 마이크로서비스 이해

▶ 마이크로서비스 성공 사례

● Line Engineering

- <https://engineering.linecorp.com/ko/blog/building-large-kubernetes-cluster/>
- 라인에서는 Kubernetes 기반 서비스 Nucleo를 자체 개발 사용
- Nucleo에는 현재 800여 개의 앱이 약 4,000개의 pod로 서비스, 계속적으로 증가(2019.08.08)

LINE Engineering

Blog

Engineering Culture

대규모 Kubernetes 클러스터 구축기

LINE C



Lee Seung | 2019.08.08

kubeadm HA topology - external etcd

