

컨테이너 교육

Oracle Korea
김종규



Confidential – Oracle Internal/Restricted/Highly Restricted

순서

시간	항목	제목
2시간	시작	1. 서버 가상화
		2. 하이퍼바이저
		3. 컨테이너
		4. 클라우드
	도커	1. 도커의 이해
		2. 도커 이미지
		3. 도커 레지스트리
		4. 도커 기본 사용법
2시간	서비스 구축	1. 웹서비스
		2. 데이터베이스
	쿠버네티스	1. 컨테이너 오케스트레이션
		2. 쿠베르네티스
		3. OKE (Oracle Container Engine for Kubernetes)
		4. 기본 사용법

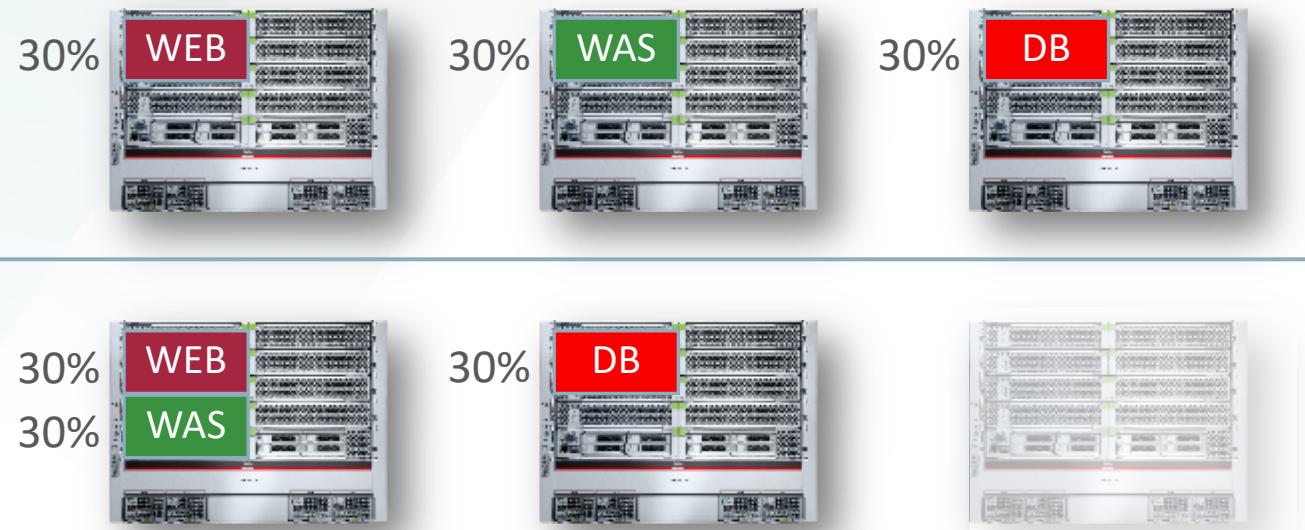
시작

- 서버 가상화
- 하이퍼바이저
- 컨테이너
- 클라우드

가상화

서버의 가상화

- 물리적인 컴퓨터 리소스의 특징을 다른 시스템, 응용 프로그램, 최종 사용자들이 리소스와 상호 작용하는 방식으로부터 감추는 기술
- 서버의 물리적인 자원을 효율적으로 사용하기 위한 기술

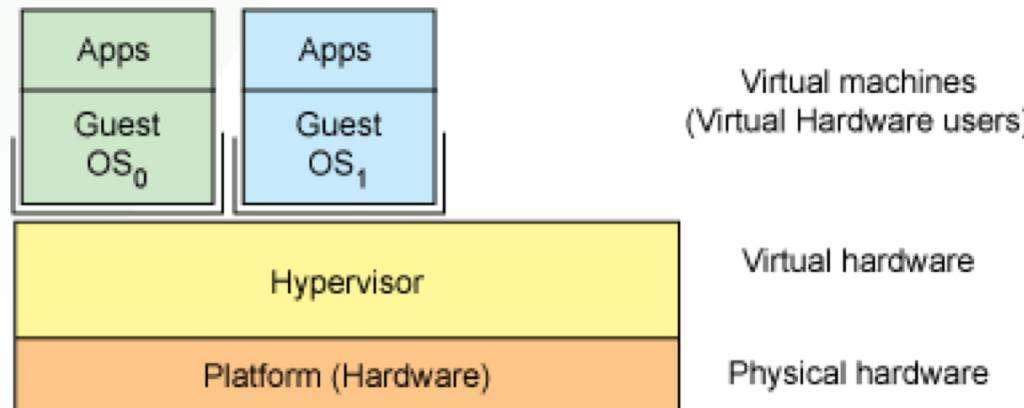


가상화와 하이퍼바이저

서버의 가상화

- 정의

- 일반적인 계층화된 아키텍처에서 플랫폼 가상화를 지원하는 계층을 하이퍼바이저 라고 하며 가상 시스템 모니터 또는 VMM(Virtual Machine Monitor)이라고도 함.
- 게스트 운영체제의 각 인스턴스를 가상시스템(VM, Virtual Machine)이라고 하며 이러한 VM에서는 가상화된 하드웨어를 전용 하드웨어로 인식

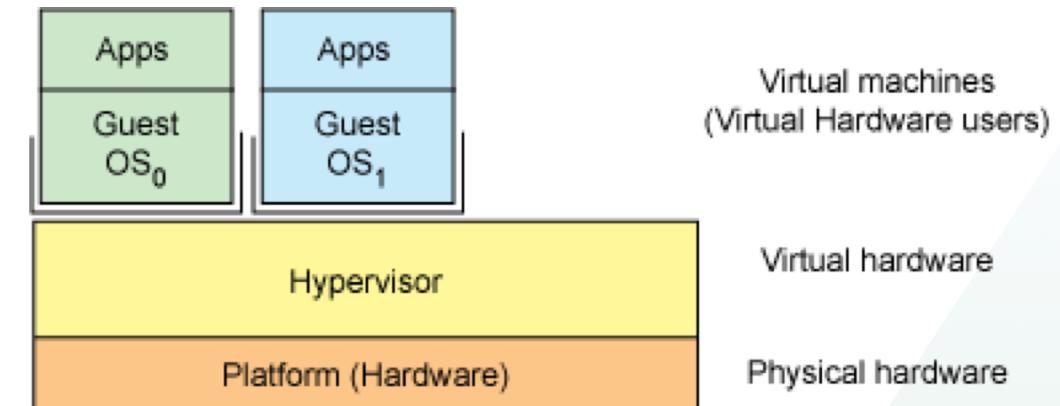


source : https://www.kdata.or.kr/info/info_04_view.html?field=&keyword=&type=techreport&page=135&dbnum=128488&mode=detail&type=techreport

하이퍼바이저 기본개념

하이퍼바이저

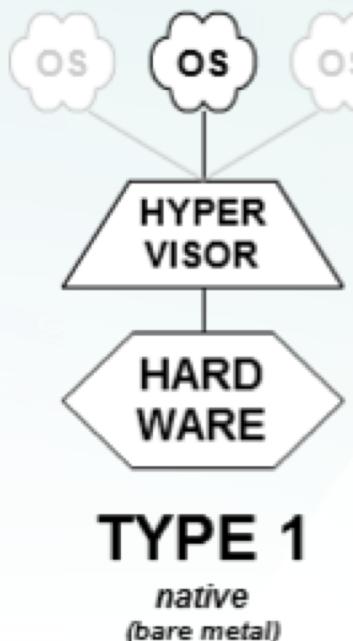
- 하이퍼바이저
 - 호스트 시스템에서 다수의 게스트OS를 돌리기 위한 플랫폼
 - 윈도우즈에서 리눅스를 설치하기 위한 VM, VirtualBox 등
- 호스트 OS
 - 물리시스템에 설치된 OS
- 게스트 OS
 - 가상머신/컨테이너 위에 설치된 OS



하이퍼바이저 유형

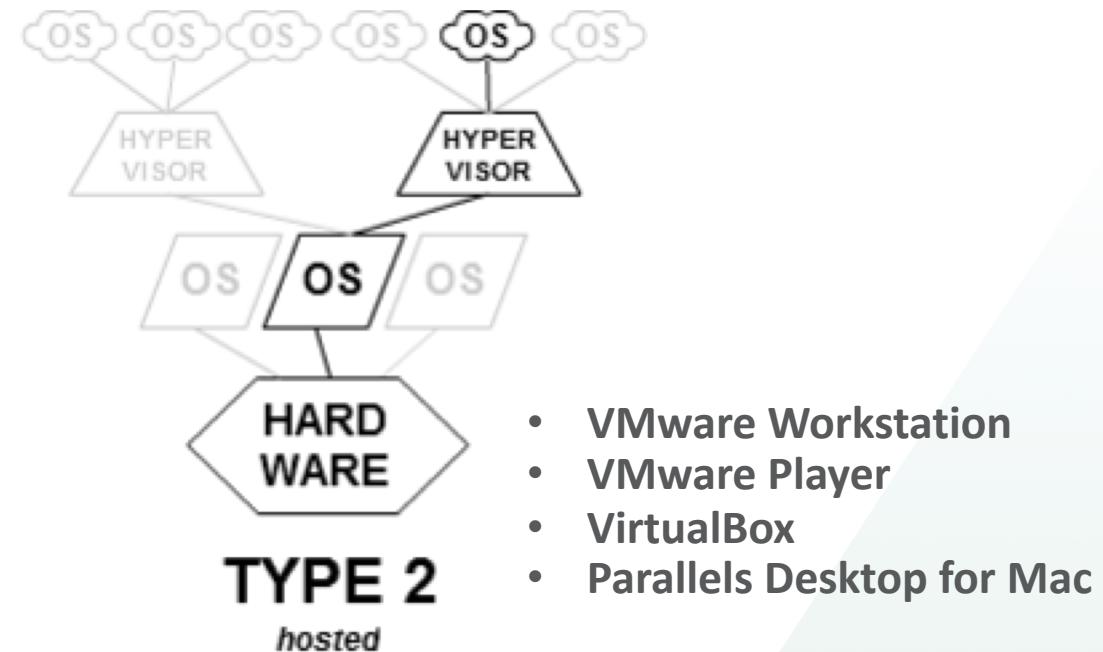
하이퍼바이저

- Type 1
 - 하이퍼바이저가 하드웨어 위에서 바로 실행



- Xen
- Oracle SPARC VM Server
- Oracle x86 VM Server
- Microsoft Hyper-V
- VMware ESX/ESX

- Type 2
 - 호스트OS 위에 하이퍼바이저를 실행시키는 방식



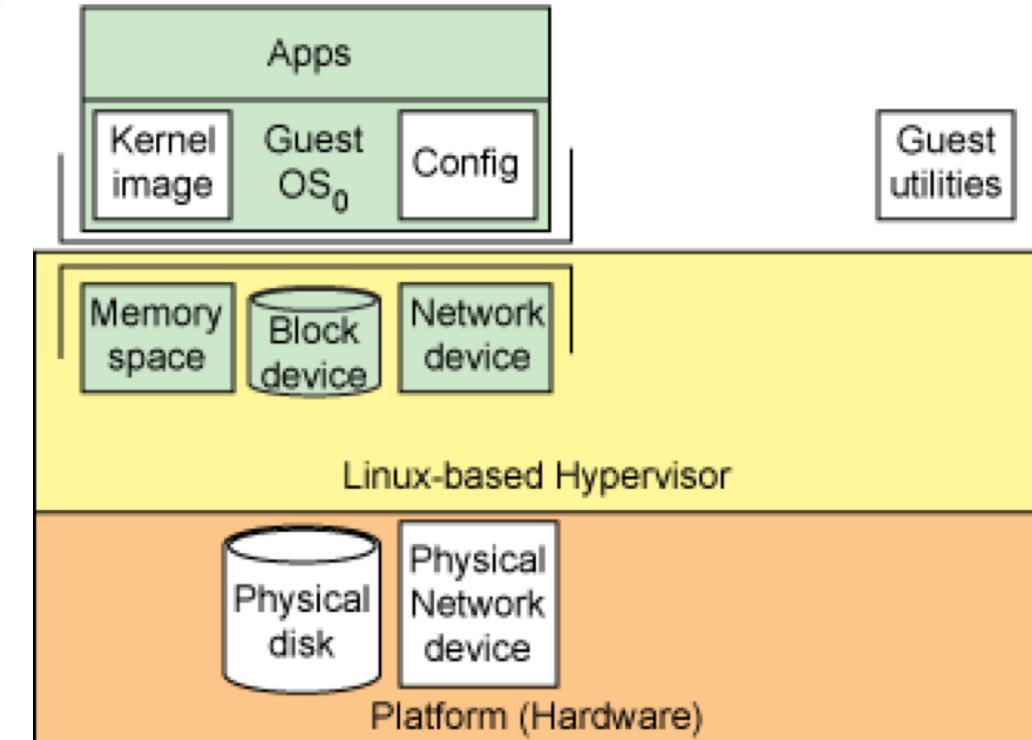
- VMware Workstation
- VMware Player
- VirtualBox
- Parallels Desktop for Mac

source : <https://en.wikipedia.org/wiki/Hypervisor>

하이퍼바이저 요소

하이퍼바이저

- 최소한의 리소스 맵핑
 - 부팅할 커널 이미지
 - 구성(사용할 IP주소 및 메모리 용량)
 - 디스크
 - 네트워크 장치
 - 게스트 도구 세트



source : https://www.kdata.or.kr/info/info_04_view.html?field=&keyword=&type=techreport&page=135&dbnum=128488&mode=detail&type=techreport

하이퍼바이저 장단점

하이퍼바이저

- 장점
 - 동일한 물리적 하드웨어에서 구동할 수 있지만 논리적으로 분리됨.
 - 한 VM에서 오류가 발생하거나 작동이 멈추거나 악성코드 공격을 받아도 다른 VM으로 확장이 되지 않음.
 - 하드웨어와 독립적이기 때문에 로컬 또는 원격 가상화 서버간 이동 가능
- 단점
 - 각 VM은 운영체제의 전체 복제본 뿐만 아니라 운영체제가 실행해야 하는 모든 하드웨어의 가상 복제본을 구동
 - 이로 인해 소모되는 RAM과 CPU가 빠른 속도로 늘어남

컨테이너

컨테이너



컨테이너 등장

컨테이너

- 하이퍼바이저의 단점 극복
 - 가상화에 항상 게스트OS를 설치하는 것이 무거움.
 - 용량이 커서 네트워크로 이미지를 받기에 부담스러움
 - OS 가상화만 주력해 배포와 관리의 부족
- 컨테이너 등장
 - OS는 리눅스를 사용, 수행은 커널을 통해서 수행(게스트OS 없음)
 - 가상화가 아닌 격리
 - 작은 이미지로 배포와 관리가 좋음
 - 하드웨어를 효율적으로 사용

VM과 컨테이너

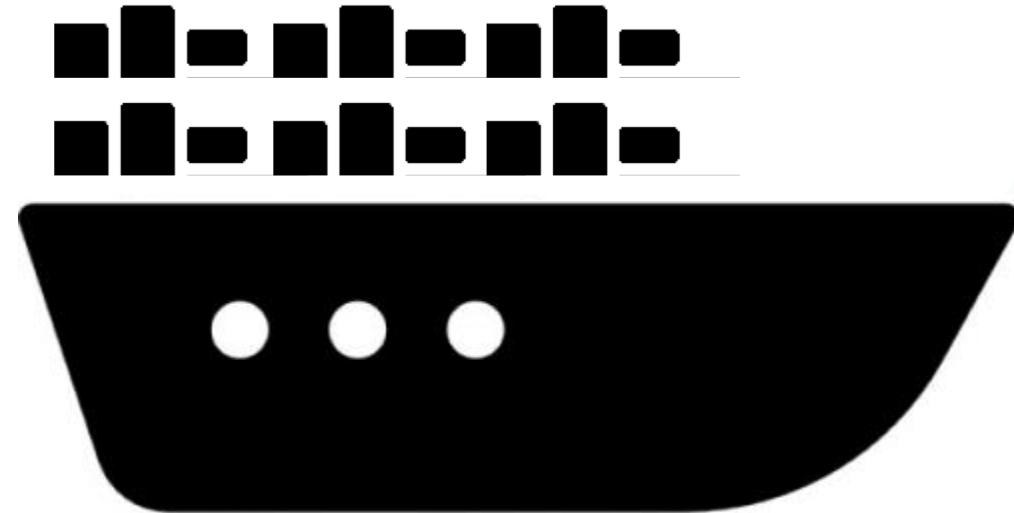
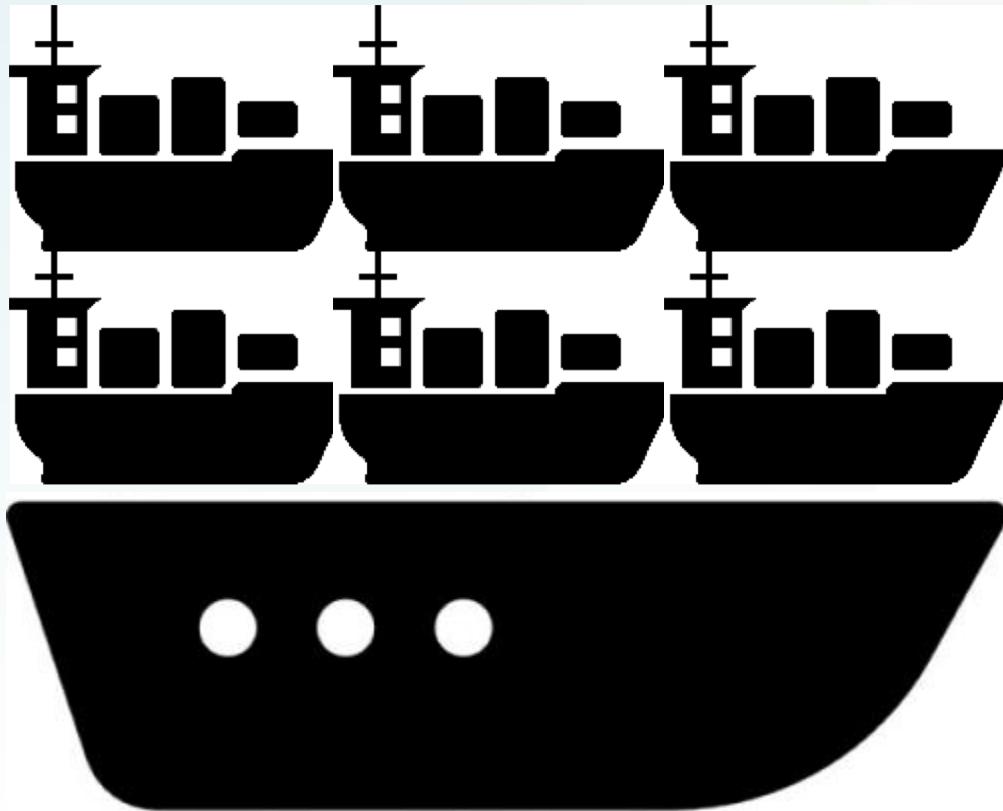
컨테이너

- 비슷한 목적
 - 애플리케이션과 그 의존성들을 독립된 단위로 묶어 격리
 - 어디서는 실행 가능하게 하는 것
 - 물리적 하드웨어의 필요성을 제거하여 컴퓨터의 자원을 에너지 면에서, 비용 면에서 더 효율적으로 사용할 수 있게 해줌.
- 다른 점
 - VM : 하드웨어의 가상화
 - 컨테이너 : 가상화가 아닌 격리

VM과 컨테이너

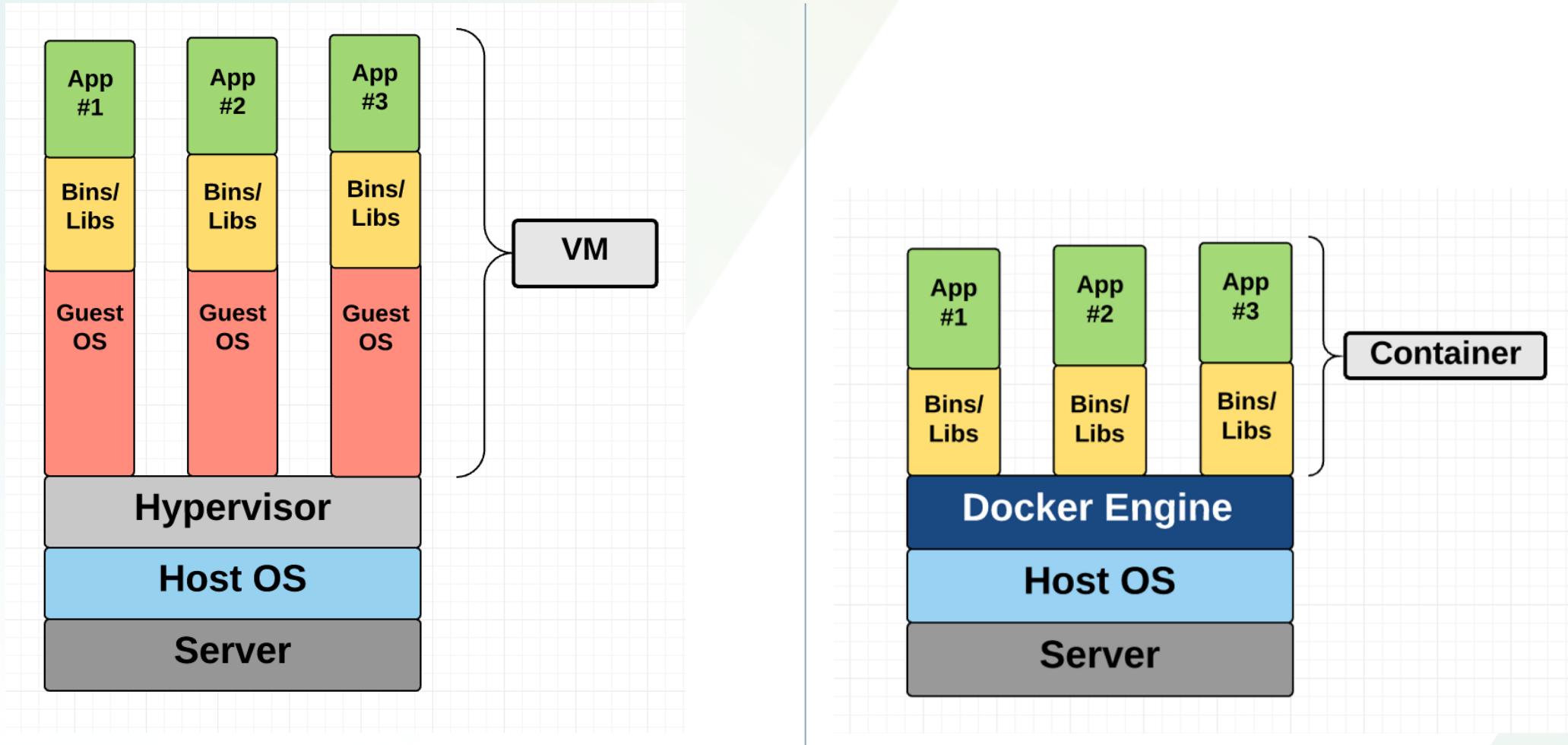
컨테이너

VM | 컨테이너



VM과 컨테이너

컨테이너



source : <https://medium.freecodecamp.org/a-beginner-friendly-introduction-to-containers-vms-and-docker-79a9e3e119b>

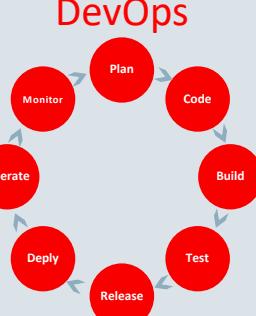
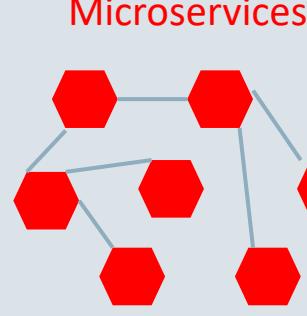
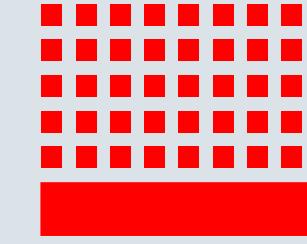
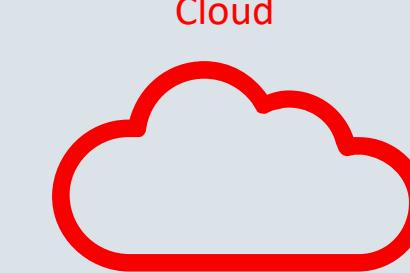
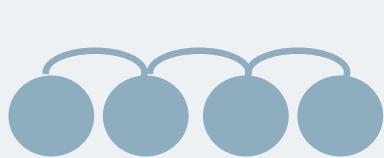
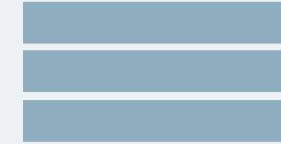
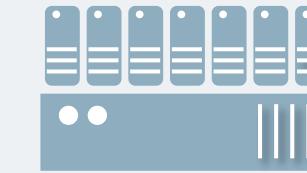
- 리눅스에서 컨테이너를 기술하기 위한 소프트웨어는 여러개
 - OpenVZ, Virtuozzo, libvirt, Docker, systemd, Warden
- 그 중 Docker Inc 회사에서 구현한 Docker는 단순한 컨테이너의 구현을 넘어 컨테이너 기반 가상화 에코시스템을 구축
- 현재는 (컨테이너 = 도커) 의 개념.
- 여러 클라우드 업체들이 기본적으로 지원





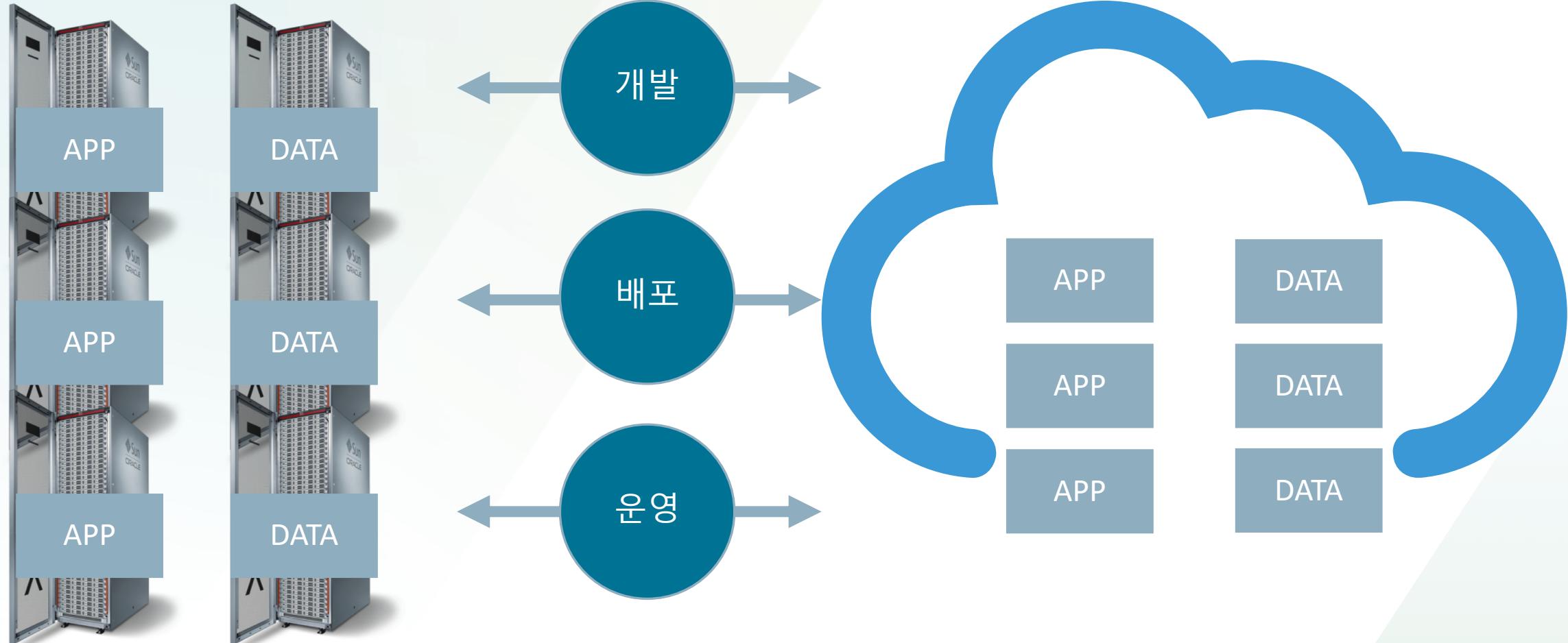
애플리케이션 개발 및 환경의 진화

클라우드

	Development Process	Application Architecture	Deployment and Packaging	Application Infrastructure
Now				
~2010				
~1980				
~1990				

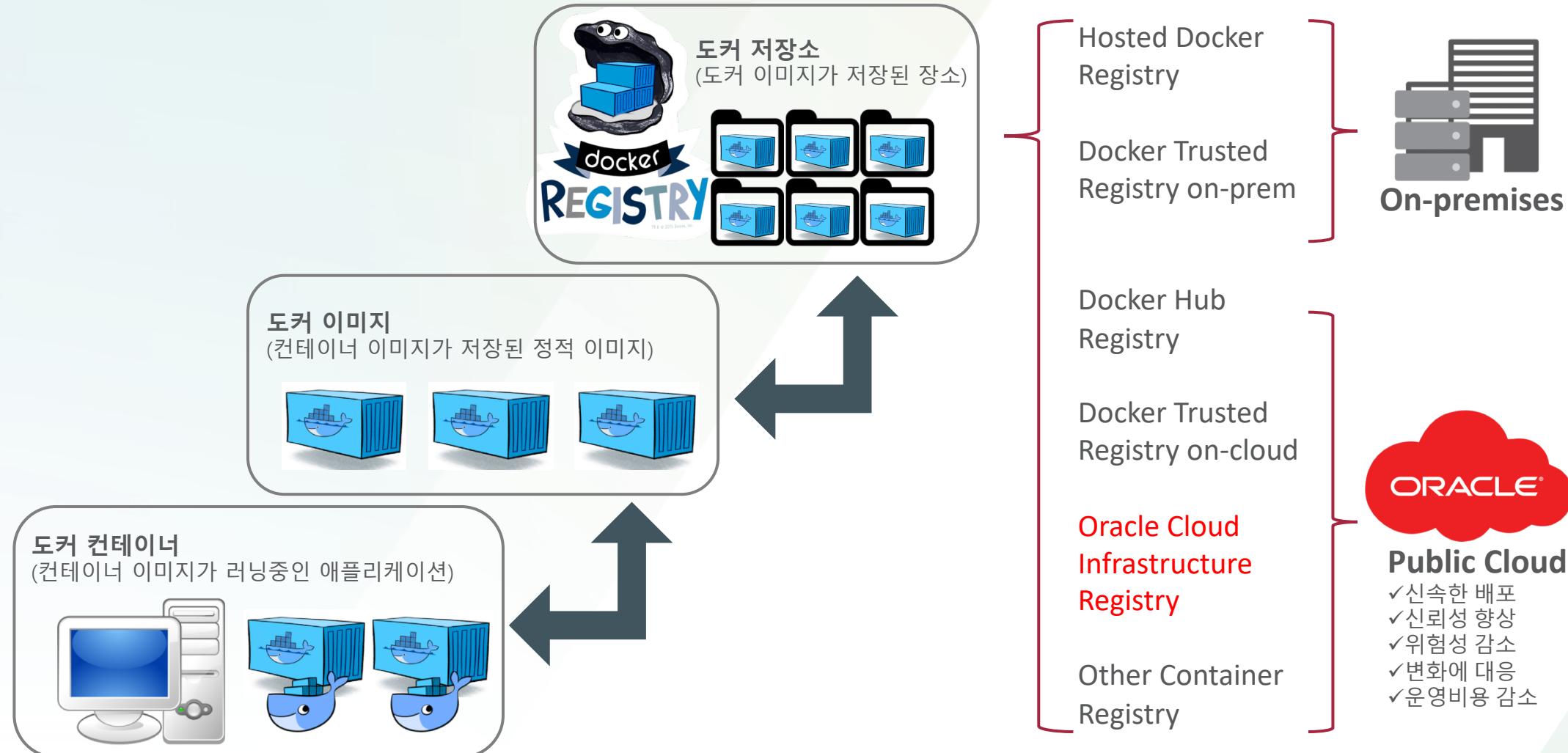
클라우드 애플리케이션

클라우드



클라우드 네이티브 애플리케이션

클라우드

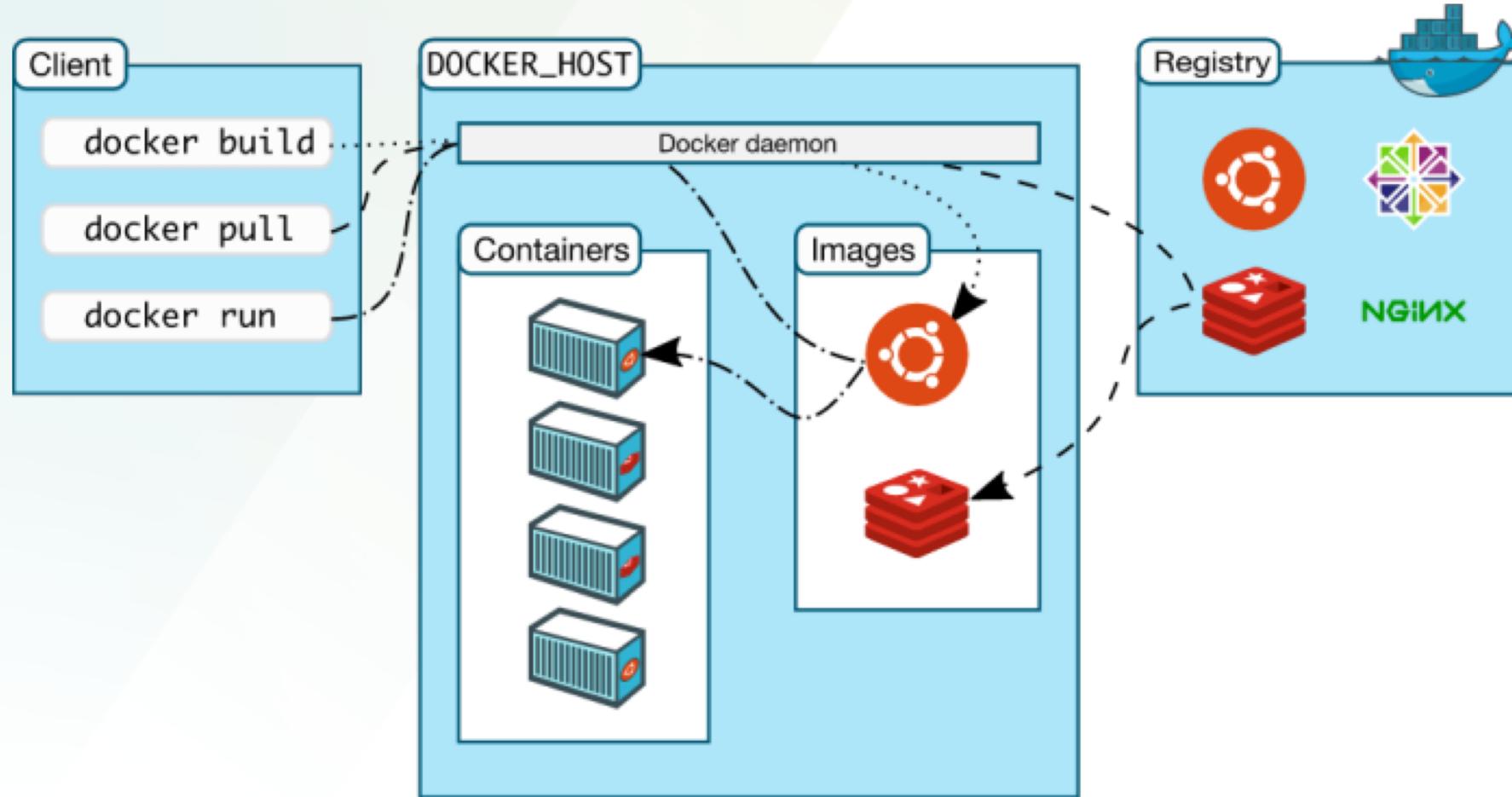


도커

- 도커의 이해
- 도커 이미지
- 도커 레지스트리
- 도커 기본 사용법

도커의 이해

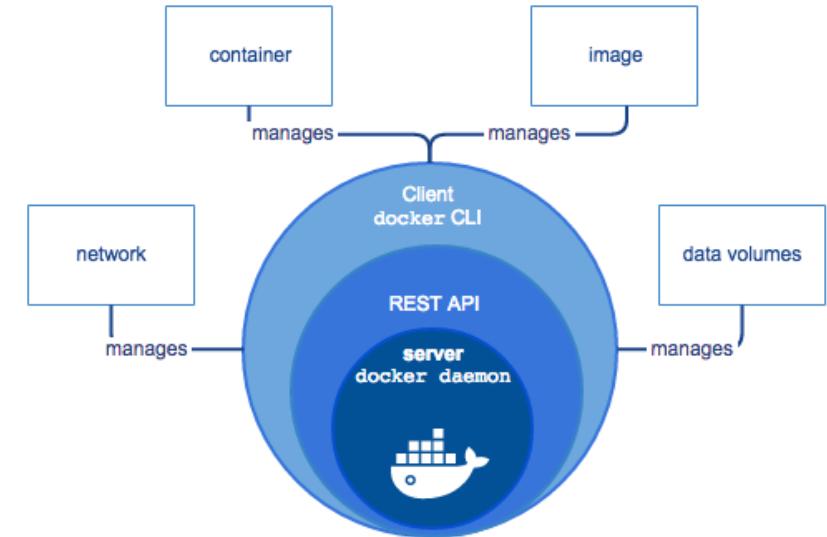
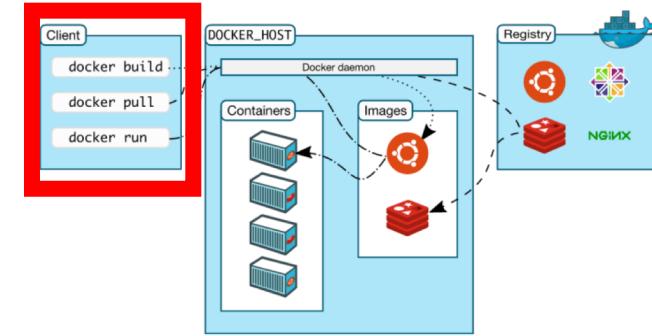
도커



도커 CLI

도커

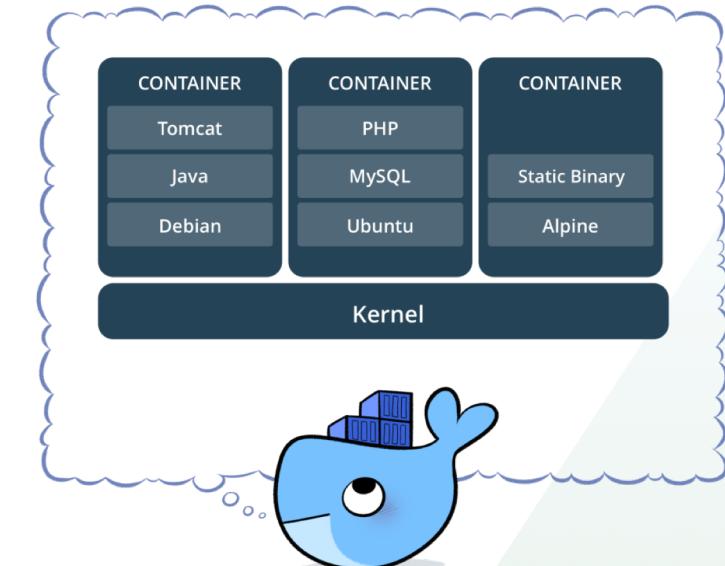
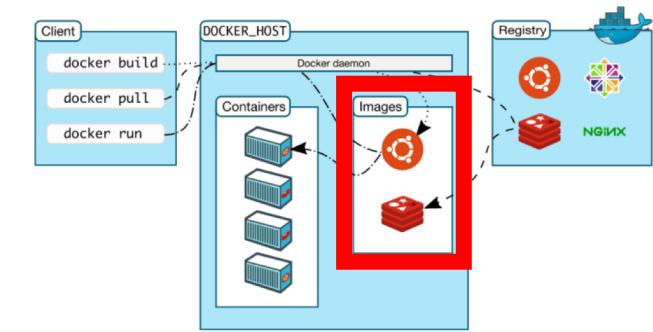
- 도커 프로그램에 포함된 커맨드 인터페이스
 - 도커에 대한 전체 명령을 수행
- docker 커맨드 [파라메터]
 - docker build
 - docker run hello-world
 - docker ps –a
 - docker images
 - docker version
 - docker rm [컨테이너]
 - docker rmi [이미지]



도커 이미지

도커

- 도커 컨테이너에서 수행될 이미지
 - 하나의 이미지가 복수개의 컨테이너에서 수행 가능
 - 도커 레지스트리에 이미지가 업로드 및 버전관리
 - 도커 호스트에 이미지가 다운로드 및 수행



도커 이미지 만들기

도커

- Dockerfile 파일
 - 도커 이미지를 만들기 위한 파일
 - 프로그램 및 이미지의 생성시의 내용을 표기
- .dockerignore 파일
 - 도커 이미지 빌드시 무시될 파일들을 기재
- docker build [이미지명] 디렉토리
 - 디렉토리 내의 Dockerfile 을 참조하여 도커 이미지를 만드는 명령어

```
FROM node:8

# 앱 디렉터리 생성
WORKDIR /usr/src/app

# 앱 의존성 설치
COPY package*.json .
RUN npm install

# 앱 소스 추가
COPY .

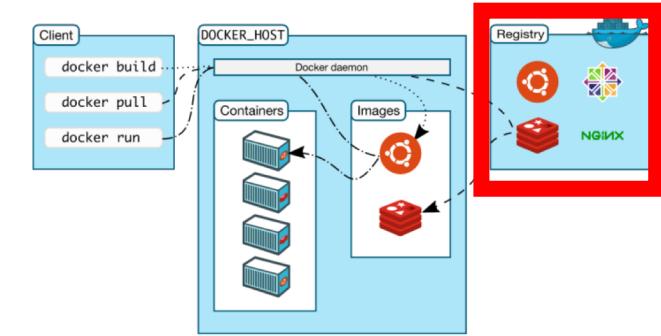
EXPOSE 8080
CMD [ "npm", "start" ]
```

도커 레지스트리

도커

- 도커 이미지를 저장하고 있는 장소
- 기본은 docker.io/library
 - <https://hub.docker.com/>에서 내용 검색 가능
- 도커 이미지의 태그명으로 레지스트리 구별
- docker run hello-world

```
Unable to find image 'hello-world:latest' locally
Trying to pull repository docker.io/library/hello-world ...
latest: Pulling from docker.io/library/hello-world
1b930d010525: Pull complete
Digest: sha256:2557e3c07ed1e38f26e389462d03ed943586f744621577a99efb77324b0fe535
Status: Downloaded newer image for hello-world:latest
aefa06e77ac9f47266549241445300f9330d74fb40f7baea4701b8cf5602db62
```



도커의 기본 사용법

도커

- 도커의 명령어
 - docker pull [이미지] / docker push [이미지]
 - docker images / docker ps –a
 - docker run / docker exec
 - docker start / docker stop
 - docker rm / docker rmi
 - docker export [컨테이너] / docker save [이미지]
 - docker import [tar] / docker load [tar]

1. VM 접속

<http://bit.ly/container-setup>

2. 도커 테스트

<http://bit.ly/container-docker>

서비스 구축

- 웹서비스
- 데이터베이스

이미지 검색

서비스 구축

- <https://hub.docker.com>
 - docker.io에 저장되어있는 도커 이미지 검색 URL
 - 검색한 도커 이미지의 상세한 설명, 버전, 태그를 보여줌.
- docker search [이미지명]
 - <https://hub.docker.com> 에서의 검색과 동일
 - 간단한 설명만 보여줌

INDEX	NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
docker.io	tomcat	Apache Tomcat is an open source implementati...	2291	[OK]	
docker.io	tomee	Apache TomEE is an all-Apache Java EE certif...	61	[OK]	
docker.io	dordoka/tomcat	Ubuntu 14.04, Oracle JDK 8 and Tomcat 8 base...	52	[OK]	
docker.io	davidcaste/alpine-tomcat	Apache Tomcat 7/8 using Oracle Java 7/8 with...	34	[OK]	
docker.io	bitnami/tomcat	Bitnami Tomcat Docker Image	27	[OK]	

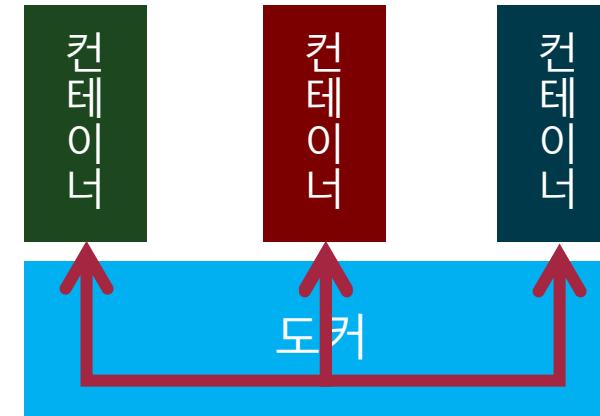
- Tomcat 검색
 - <https://hub.docker.com>에서 “tomcat”을 입력
- docker run -it --rm -p 8080:8080 tomcat:8.0
 - run options
 - -it : 컨테이너와의 상호작용을 터미널과 동일한 환경으로 설정
 - --rm : 컨테이너가 종료되면 모든 정보가 사라짐
 - -p : 호스트의 포트와 컨테이너의 포트로 포워딩
 - tomcat:8.0
 - tomcat이라는 이미지명
 - 8.0 태그로 명명된 이미지

- mysql 검색
 - <https://hub.docker.com>에서 “mysql”을 입력
- docker run --name \${USER}_mysql -e
 MYSQL_ROOT_PASSWORD=Welcome1 -d mysql:5.7
 - run option
 - --name \${USER}_mysql : 컨테이너의 이름을 지정
 - -e : 환경변수를 설정
 - -d : 컨테이너를 백그라운드로 실행
- docker exec -it \${USER}_mysql bash
 - mysql -u root --password=Welcome1

서비스 연결

서비스 구축

- 컨테이너간의 연결 (단일 호스트)

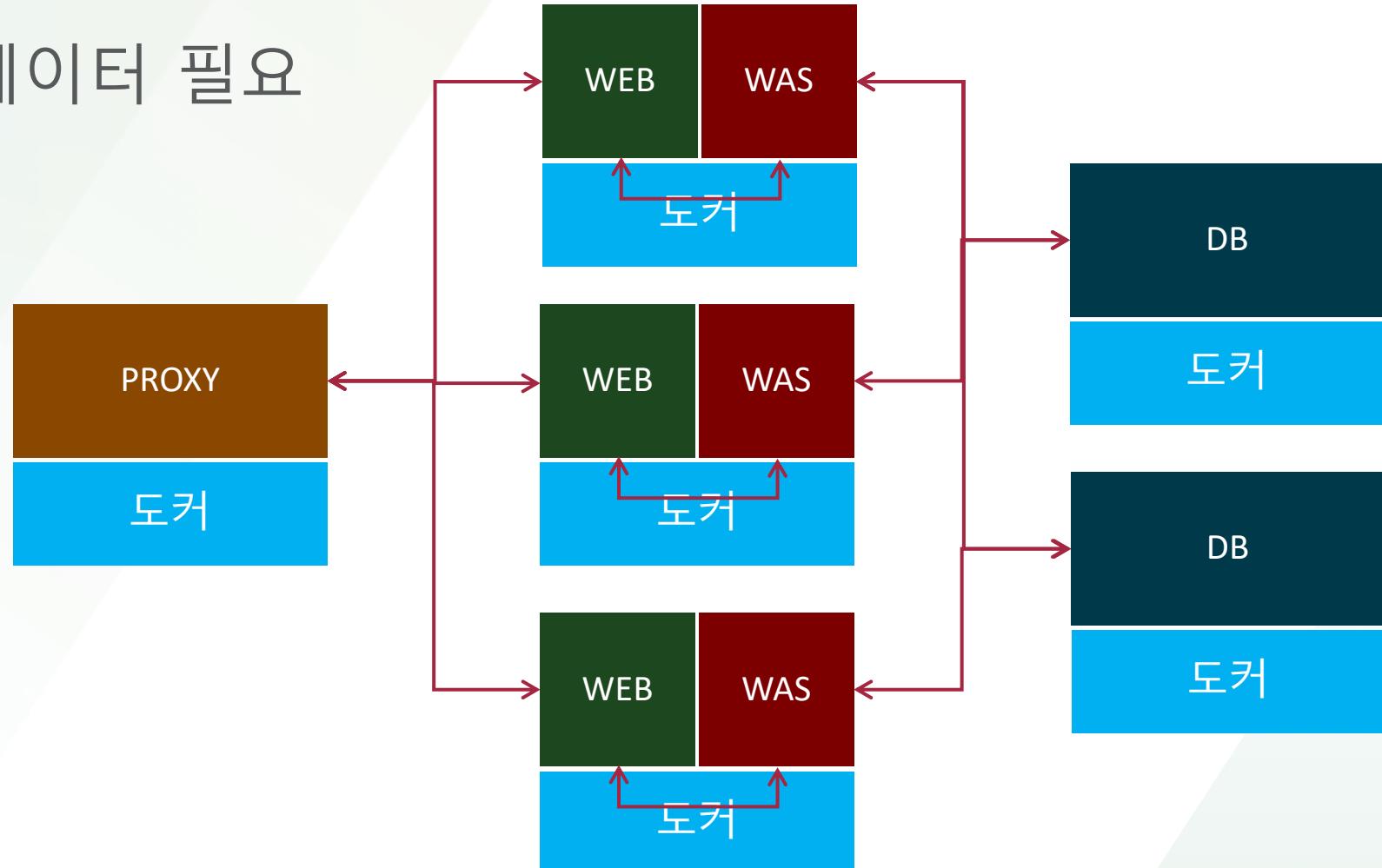


- docker run --name mydb -d db
- docker run --name mywas **--link mydb:db** was
- docker run --name myweb **--link mywas:was** web

멀티 도커 호스트 연결

서비스 구축

- 컨테이너 오케스트레이터 필요
 - Docker Swarm
 - Apache Mesos
 - Kubernetes



1. VM 접속

<http://bit.ly/container-setup>

2. 애플리케이션 테스트

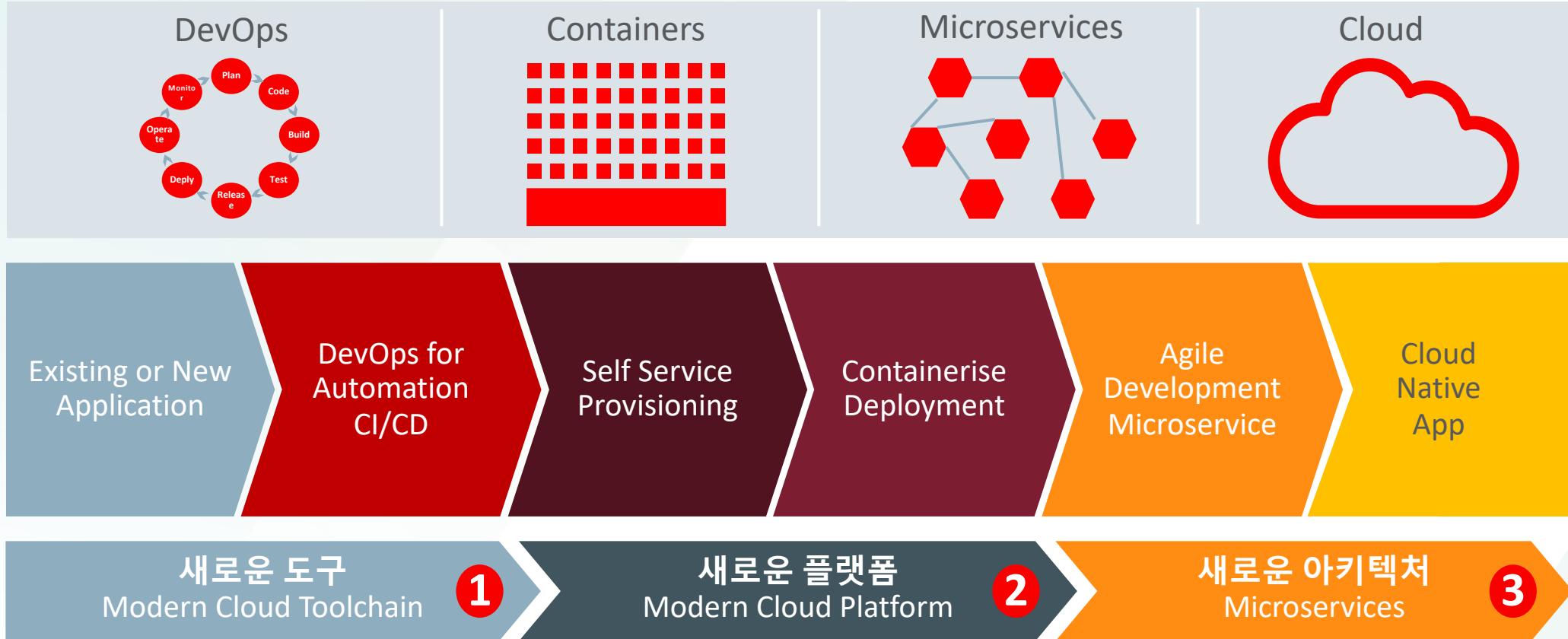
<http://bit.ly/container-app>

쿠버네티스

- 컨테이너 오케스트레이션
- 쿠베르네티스
- OKE (Oracle Container Engine for Kubernetes)
- 기본 사용법

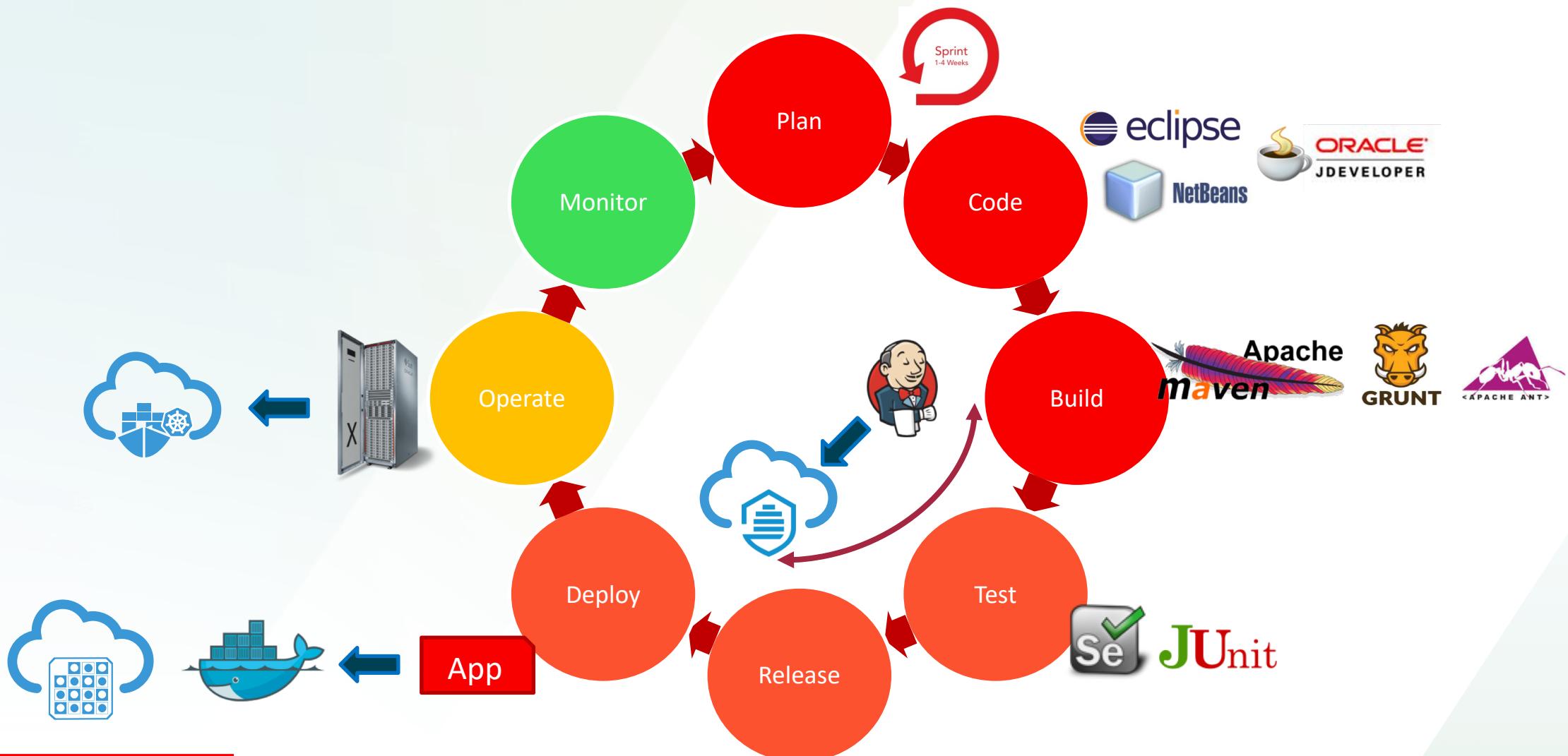
패러다임의 변화

쿠버네티스



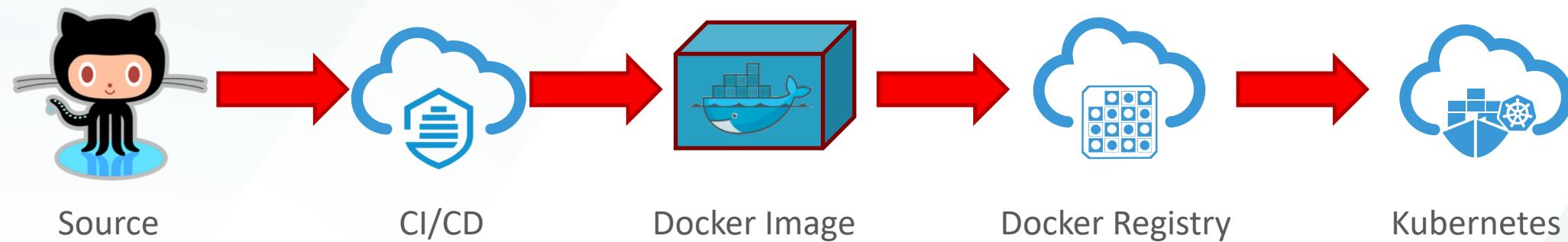
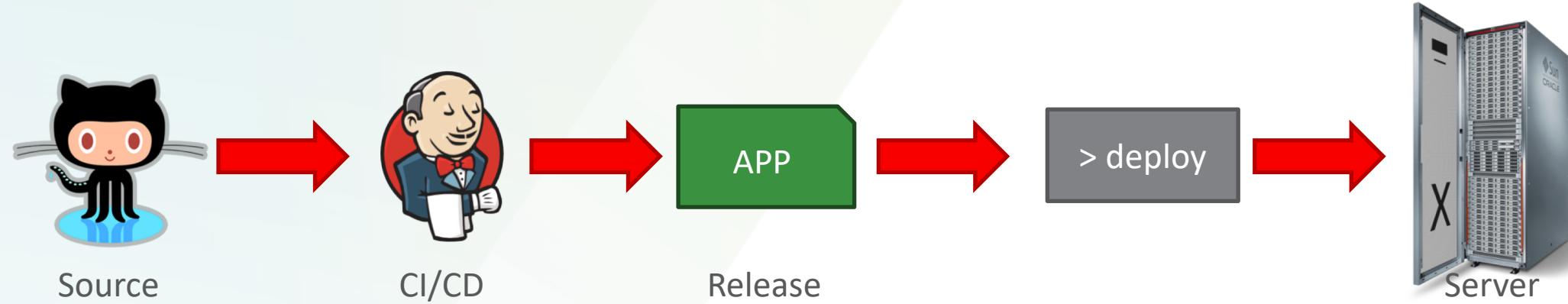
애플리케이션 개발 관리

쿠버네티스



Container Native Application Development

쿠버네티스



도커와 쿠버네티스

쿠버네티스



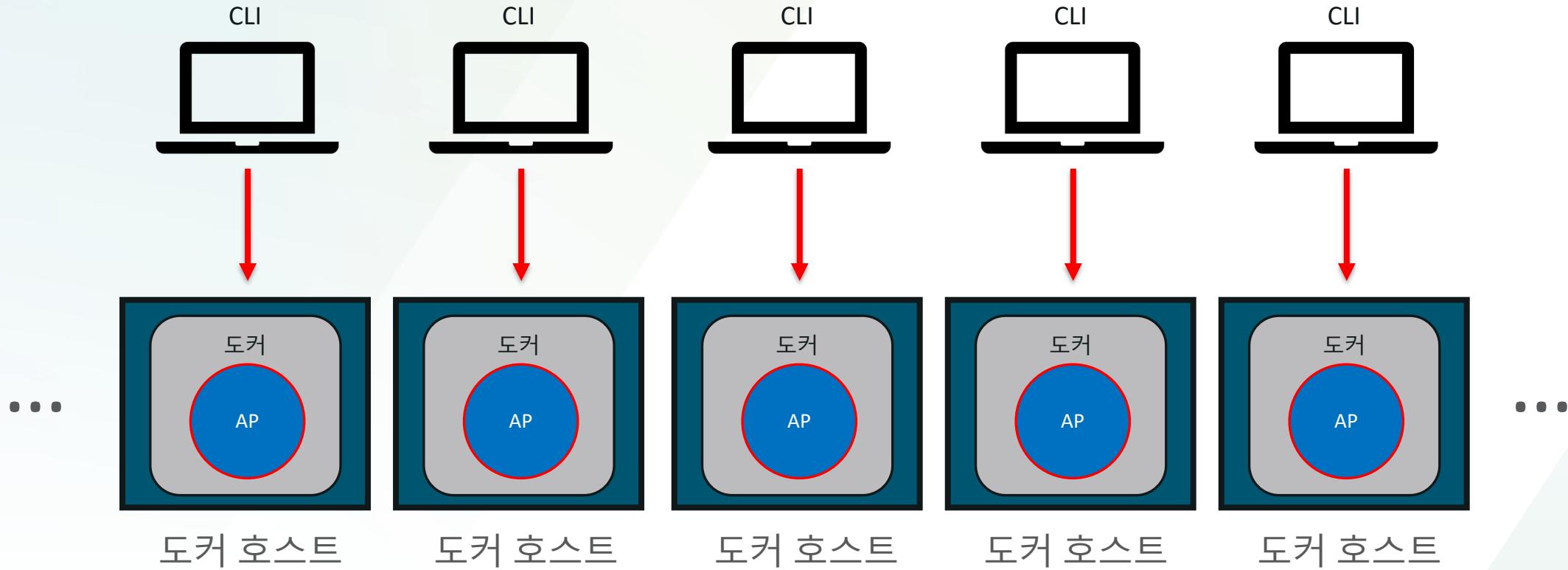
- 개발자 생산성 향상을 목표로 하는 인기 있고 사용하기 쉬운 컨테이너 툴
- 사실상 표준 컨테이너 런타임 및 이미지 형식
- 개발자 온 보딩 및 1 세대 애플리케이션에 사용



- DevOps 및 운영을 대상으로 한 광범위한 생산 관리 컨테이너 관리
- 클라우드 스케일 어플리케이션을 지원하는 복잡하지만 강력한 툴 세트
- 풍부한 운영 기능 세트, 자동 확장, 롤링 업그레이드, 상태 저장 응용 프로그램 등

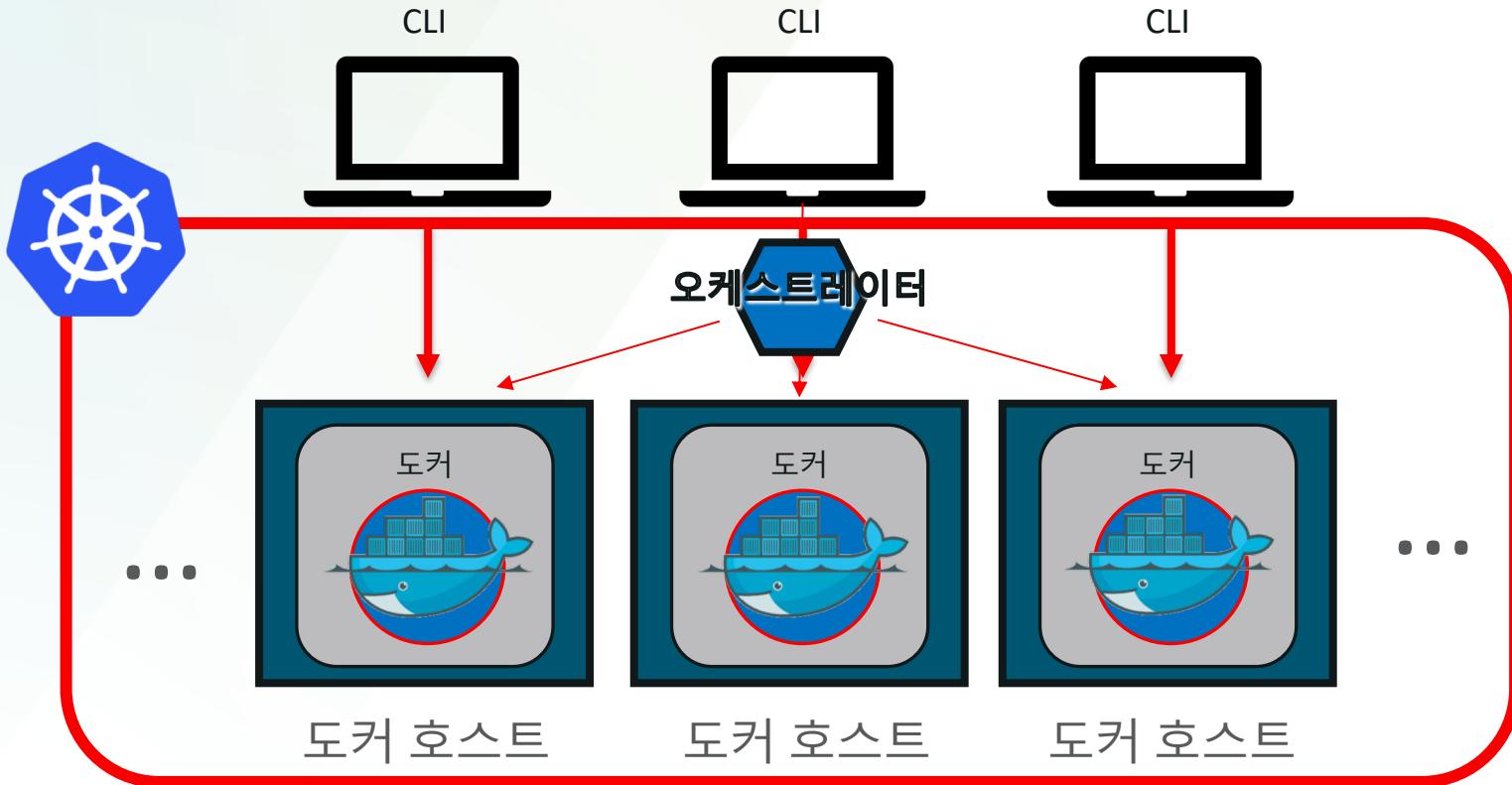
컨테이너와 오케스트레이션

클라우드



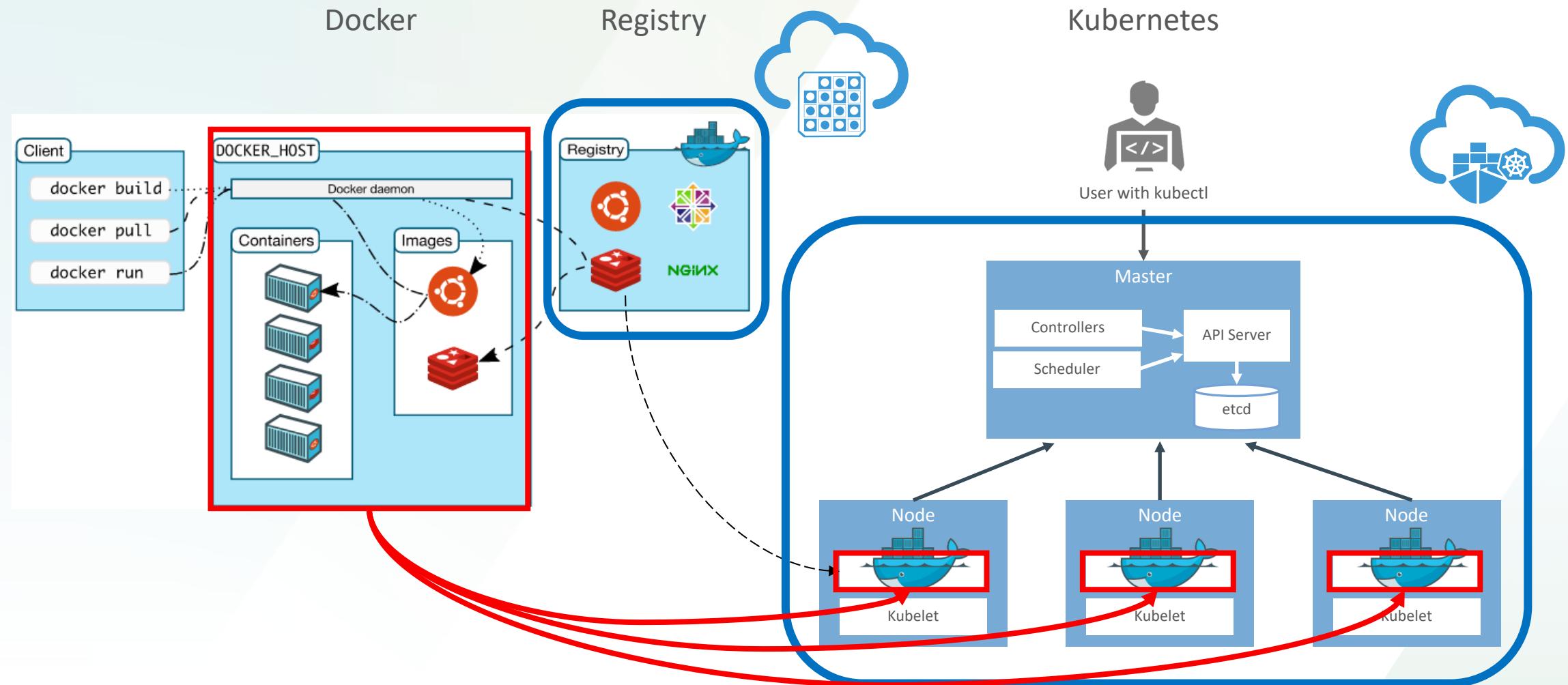
컨테이너와 오케스트레이션

클라우드



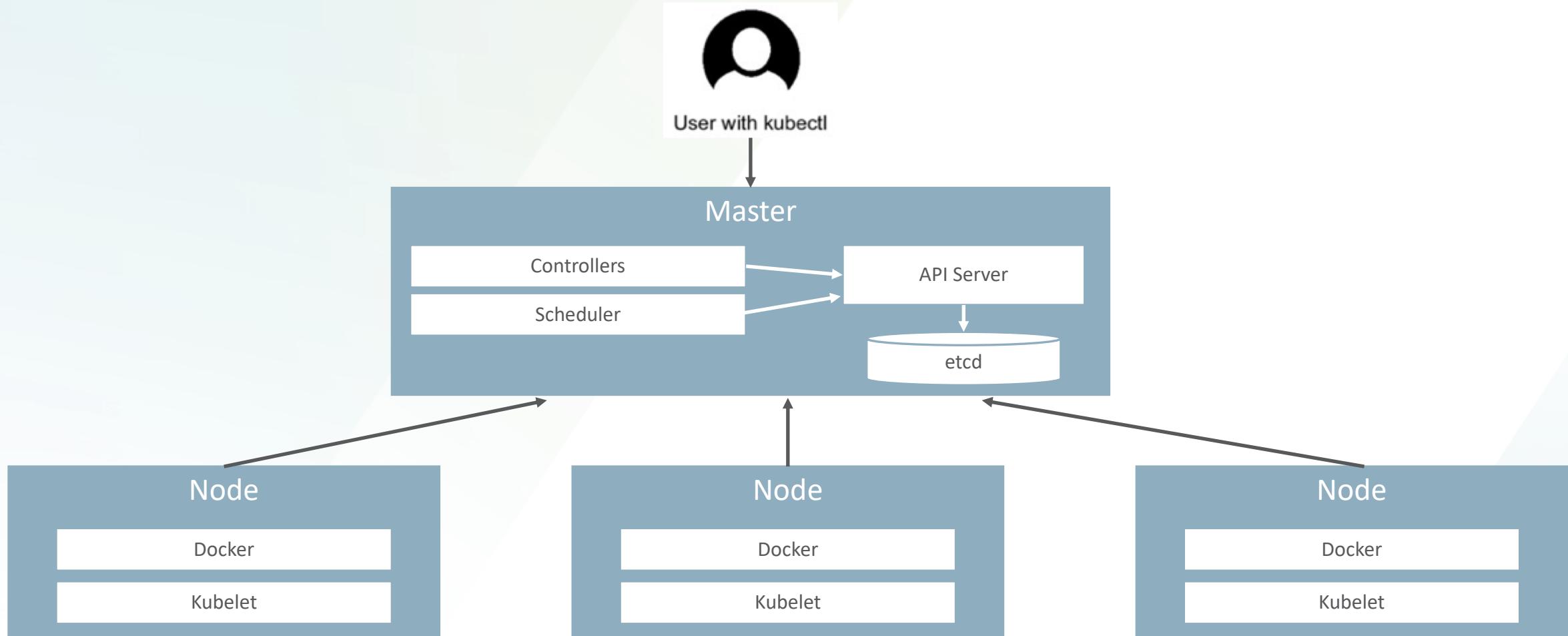
Docker / Docker Registry / Kubernetes

쿠버네티스



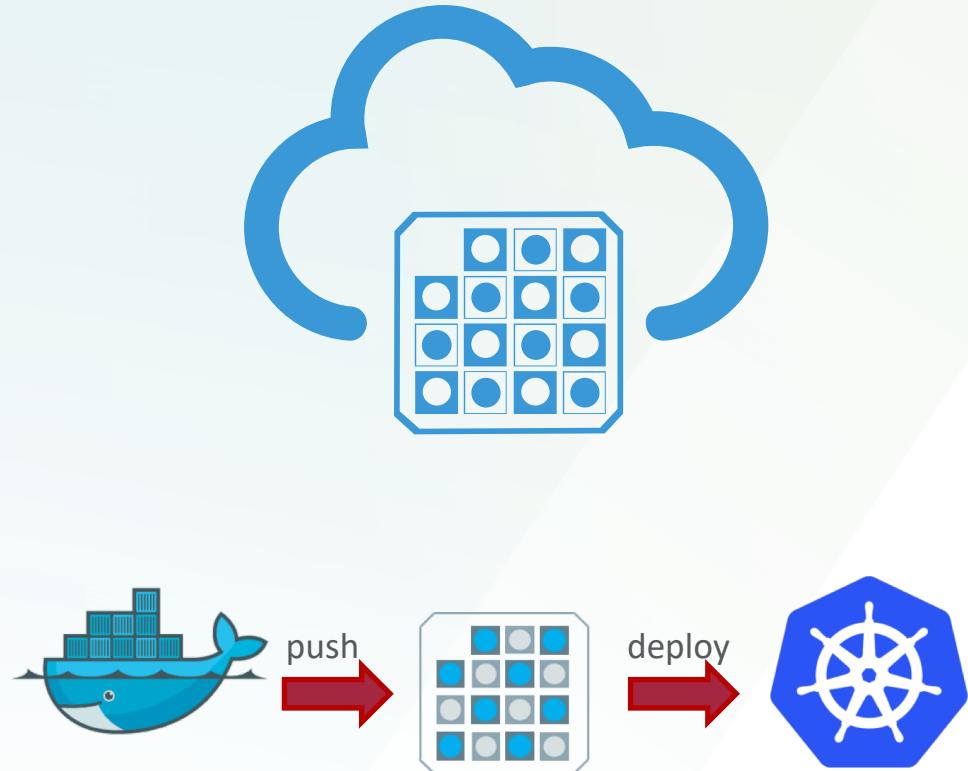
쿠버네티스 아키텍처

쿠버네티스



OCIR(Oracle Cloud Infrastructure Registry)

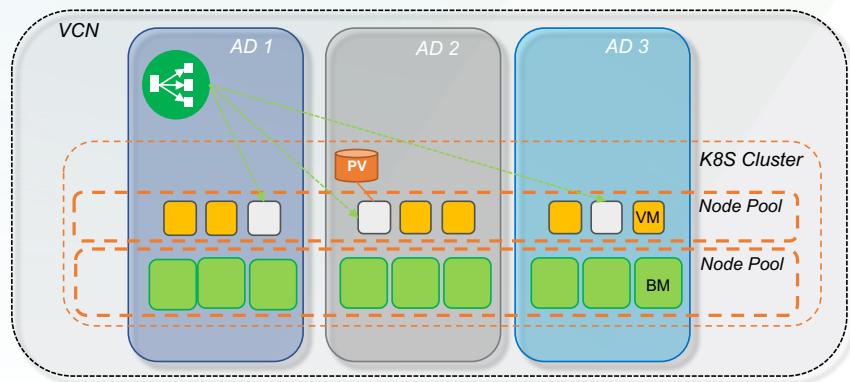
쿠버네티스



- Oracle Cloud Infrastructure의 고 가용성 Docker v2 컨테이너 레지스트리 클라우드 서비스
- Wercker Container Pipelines 및 Container Engine (OKE)과의 완벽한 통합
- 개인 저장소에 Docker 이미지 저장
- 자동 이미지 중복 제거
- Wercker를 통한 팀 접근 및 통제 관리
- 지연 없는 배포를 위한 OKE와 공동 지역 설치

OKE(Oracle Container Engine for Kubernetes)

쿠버네티스



- 컨테이너 기반의 앱을 배포하고 실행하는 관리 형 Kubernetes 컨테이너 서비스
- 표준 Kubernetes 클러스터를 즉시 생성, 확장, 관리 및 제어 할 수 있는 도구제공
- HELM 및 DNS 제공과 엔터프라이즈 애플리케이션 지원
- 개발자가 컨테이너를 신속하게 시작하고 배포 할 수 있음.
- 제어, 보안, IAM 및 오라클 차세대 클라우드 인프라 스트럭처의 예측 가능한 성능을 결합

쿠버네티스 구성방안

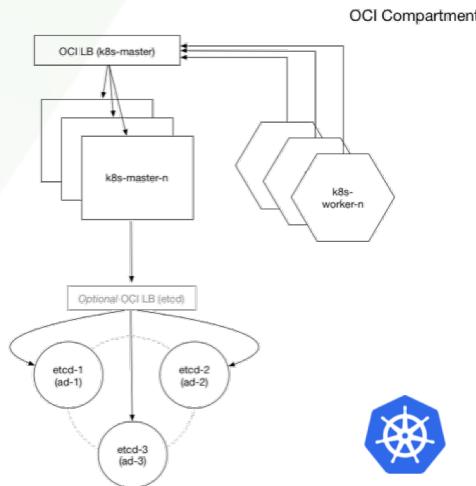
쿠버네티스

고객이 직접설치



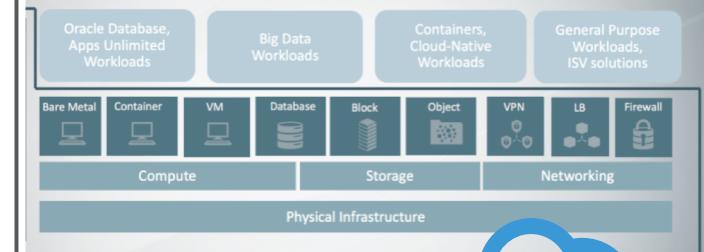
모든 영역을 고객이 관리

자동화 툴로 설치



직접 툴로 관리하는
Managed Kubernetes Service

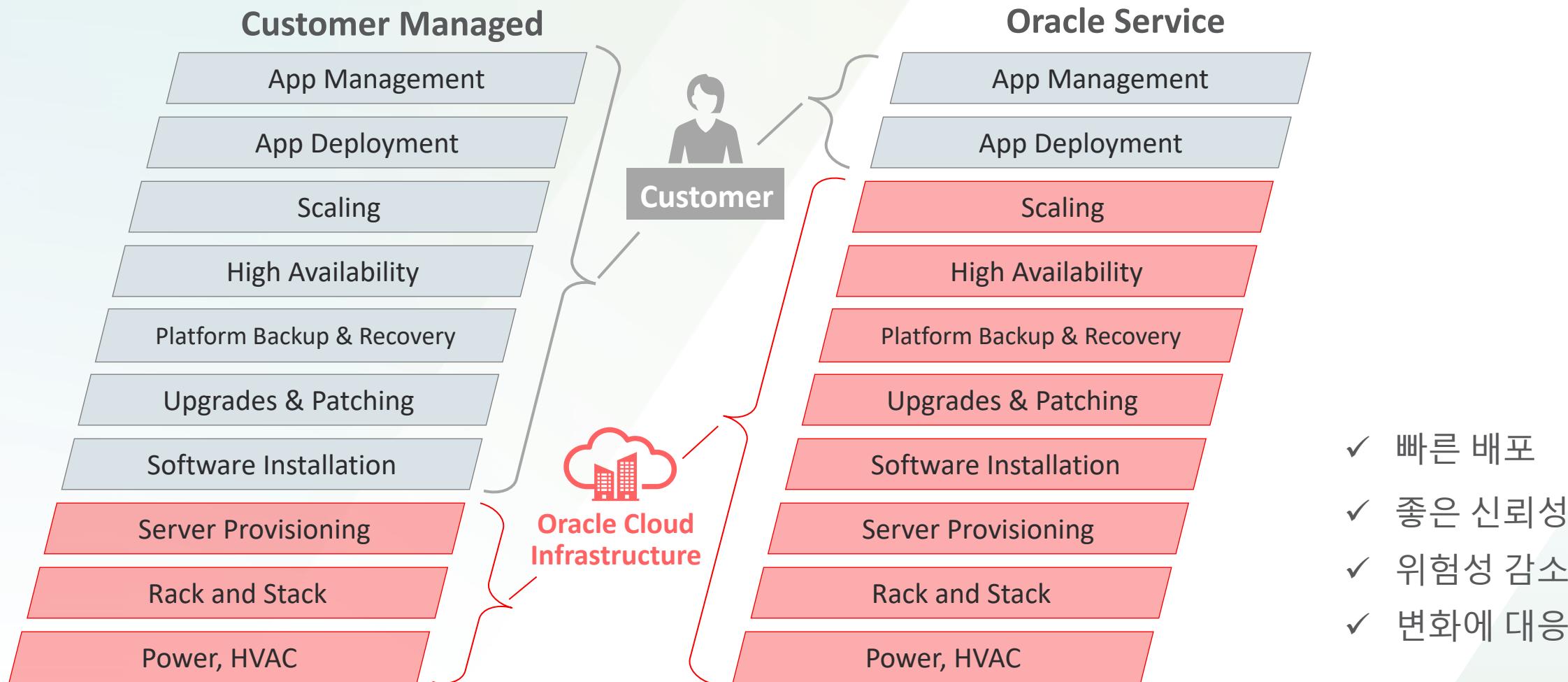
OKE(Oracle Container Engine
for Kubernetes) 사용



엔터프라이즈 클래스
Managed Kubernetes Service

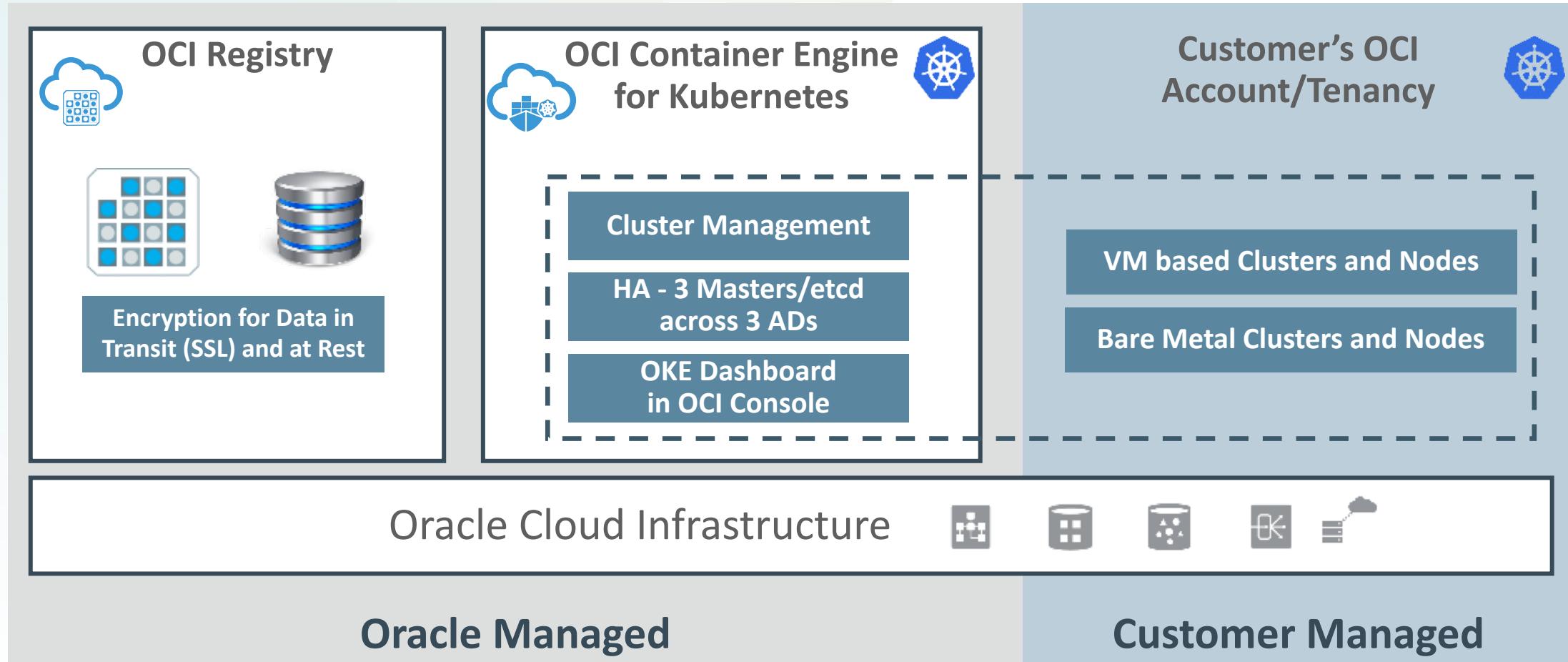
OKE의 이점

쿠버네티스



OKE와 OCIR로 작업하기

쿠버네티스



OKE Cluster 생성

쿠버네티스

<https://kubernetes.io/ko/docs/setup/>

▼ kubeadm으로 클러스터 부트스트래핑 하기

- ▶ Installing kubeadm (EN)
- ▶ Creating a single master cluster with kubeadm (EN)
- ▶ Customizing control plane configuration with kubeadm (EN)
- ▶ Options for Highly Available Topology (EN)
- ▶ Creating Highly Available Clusters with kubeadm (EN)
- ▶ Set up a High Availability etcd cluster with kubeadm (EN)
- ▶ Configuring each kubelet in your cluster using kubeadm (EN)

using kubeadm (EN)

using kubeadm (EN)

하기

with

igu

ole

cluster

ir cluster

ster

QUICK CREATE

Quickly create a cluster with default settings, also creates a dedicated network

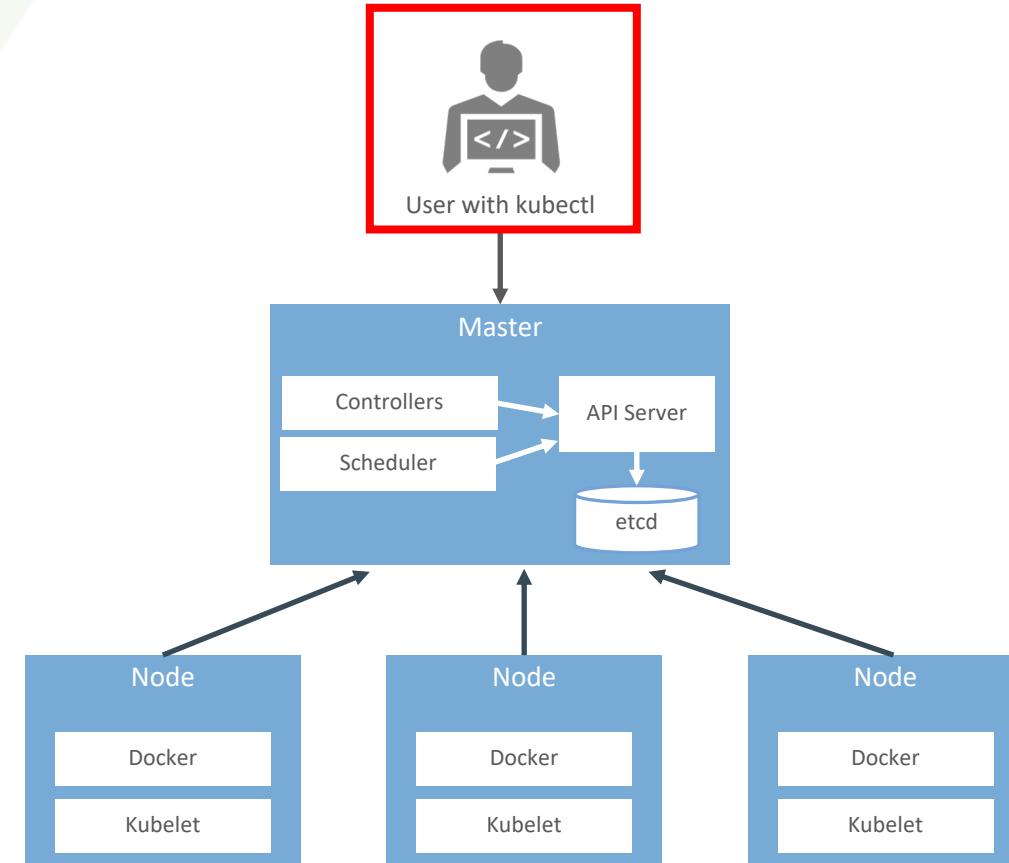
CUSTOM CREATE

Create a cluster with custom settings, assumes an existing network

기본 사용법

쿠버네티스

- kubectl
 - kubectl cluster-info
 - kubectl run
 - kubectl get
 - kubectl create
 - kubectl delete
 - kubectl scale
 - kubectl proxy



1. VM 접속

<http://bit.ly/container-setup>

2. 쿠베르네티스 테스트

<http://bit.ly/container-k8s>

ORACLE®