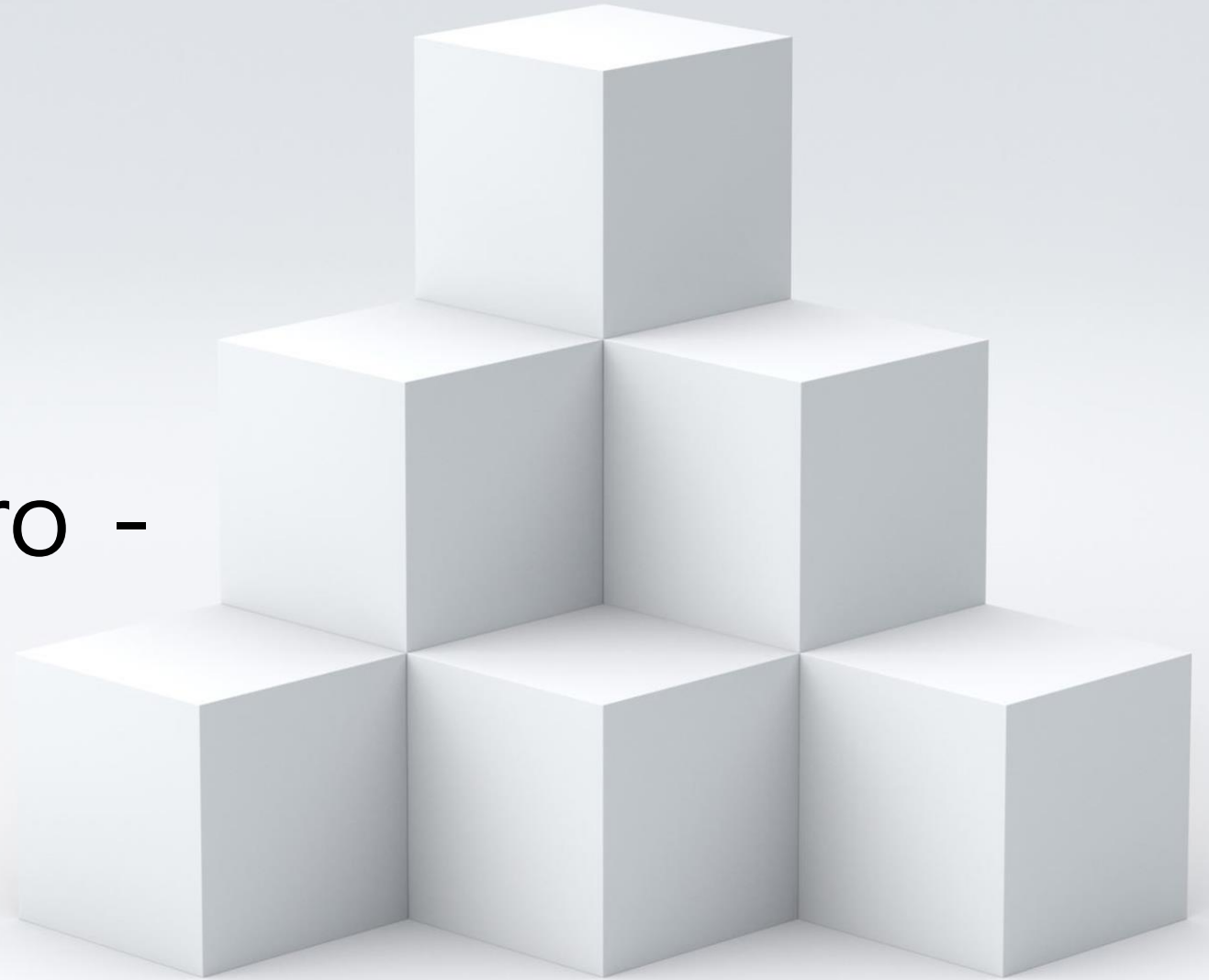


Software Engineering

- Kubernetes intro -

Professor Han-gyoo Kim

2022



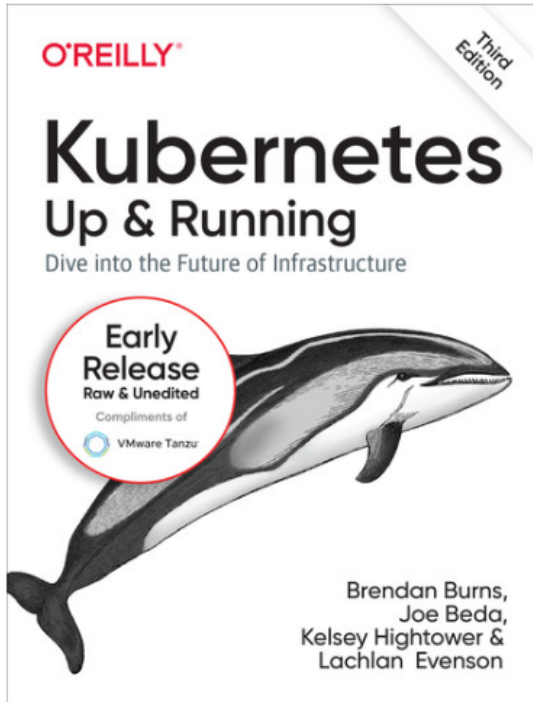
- 공식 사이트 - <https://kubernetes.io/ko/docs/home/>

 BOOK

Kubernetes: Up and Running, 3rd Edition

Write the [first review](#)

By [Brendan Burns](#), [Joe Beda](#), [Kelsey Hightower](#), [Lachlan Evenson](#)



TIME TO COMPLETE:

7h 33m

TOPICS:

[Kubernetes](#)

PUBLISHED BY:

[O'Reilly Media, Inc.](#)

PUBLICATION DATE:

August 2022

PRINT LENGTH:

320 pages

Continue

Docker Swarm vs Kubernetes(K8s)

“container orchestration tool”

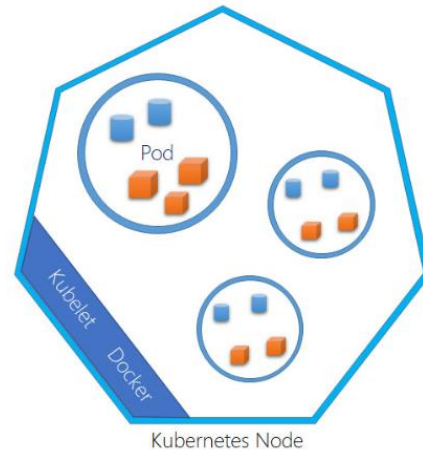
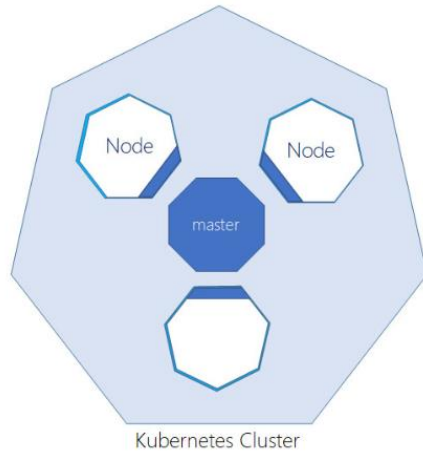
	Swarm	K8s
Container	docker container	다른 container 도 가능
Scaling	manual	fully automatic
Load Balancing	primitive	configurable
Installation	easy & fast	more time consuming
Scalability	limited	strong
Storage Volume	sharable over any container	within a pod
GUI	not available	available

Key concepts in K8s

- Pods, or groups of containers - group together container images developed by different teams into a single deployable unit
 - logical host of containers sharing network and storage resources for a service
 - containers in a pod are always co-located and co-scheduled, and run in a shared context
- Services - provide load balancing, naming, and discovery to isolate one microservice from another
- Namespaces - provide isolation and access control, so that each microservice can control the degree to which other services interact with it
- Ingress objects - provide an easy-to-use frontend that can combine multiple microservices into a single externalized API surface area

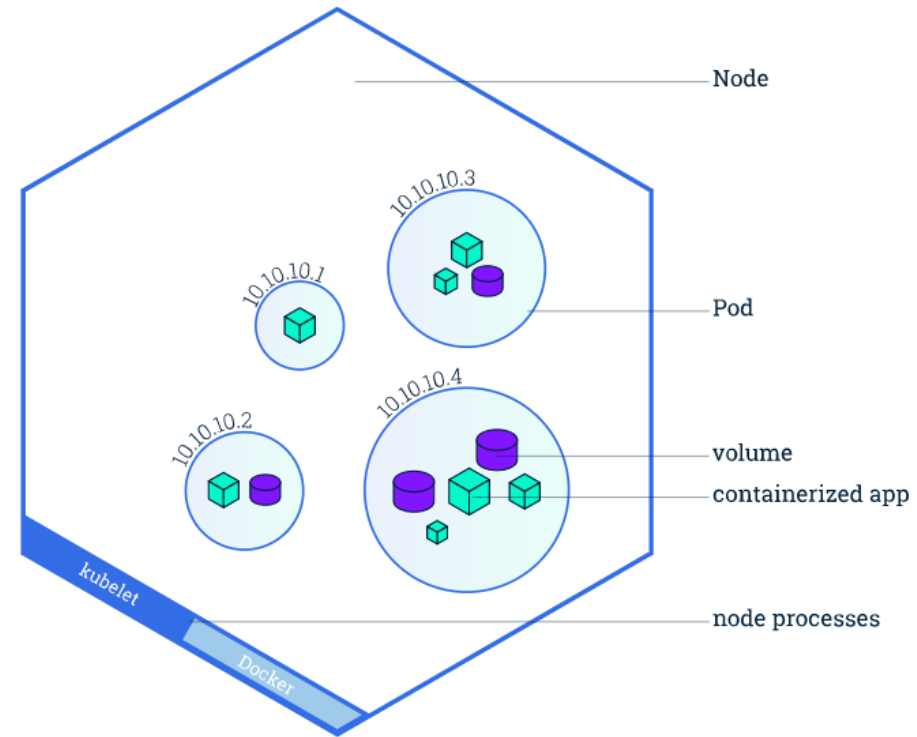
Service architecture of K8s

- K8s cluster = master node(s) + worker nodes



<https://cloudaffaire.com/what-is-pod-in-Kubernetes/>

- K8s allows to mount a storage (local, cloud, network storage) for each pod



<https://kubernetes.io/ko/docs/tutorials/kubernetes-basics/explore/explore-intro/>

Features of K8s

- (1) Automatic bin packing (RAM-aware performance based)
- (2) Service discovery & load balancing (DNS name for each service)
- (3) Storage orchestration
- (4) Self healing (restart failed containers)
- (5) Automated rollouts & rollbacks
- (6) Secret and configuration management (secret & config map in ETCD)
- (7) Batch execution (run to completion)
- (8) Horizontal scaling (CLI, UI, automatic based on CPU usage)

<https://kubernetes.io/docs/setup/best-practices/cluster-large/>

(v1.18, 2020 4월 현재)

하나의 클러스터에는

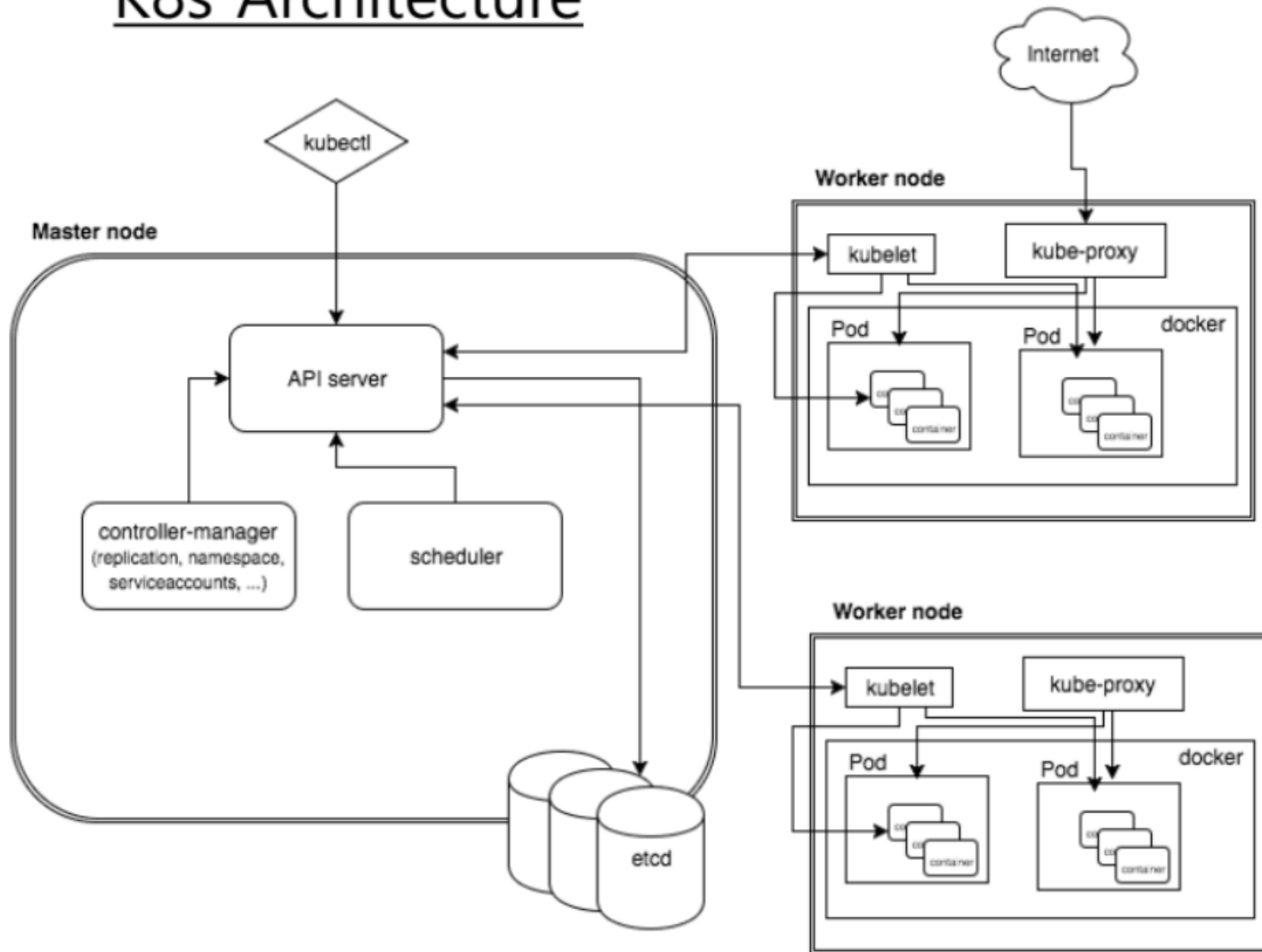
- 최대 5000 개까지의 node

- 최대 150,000 개까지의 pod

- 최대 300,000 개까지의 container

 - node 당 최대 100 개까지의 pod

K8s Architecture



master components

- API server
- scheduler
- control manager
- etcd

worker components

- kubelet
 - Container running을 책임
- Kube-proxy
 - 각 node의 network proxy
- Pod
 - 1개 이상의 docker container 포함

K8s implementations

- <https://kubernetes.io/ko/docs/home/>
- K8s labs via browser - Use k8s without installing k8s – k8s playground, play with k8s, play with k8s classroom
- Installing Kubernetes on a Public Cloud Provider (유료)
 - GCP – GKE
 - Azure
 - AWS -EKS
- Installing Kubernetes Locally
 - **kubeadm** – for bare Linux machine - a tool to build Kubernetes clusters
 - minikube (제한적)
- Running Kubernetes in Docker
 - Recent project
 - KIND – K8s in Docker
- Running Kubernetes on Raspberry Pi
- The official Kubernetes client is kubectl: a command-line tool for interacting with the Kubernetes API

Kubernetes (K8s)

Official def: Container orchestration tool

= 분산 운영체제 with no distributed file system, no cache management, no system calls, no per process control, etc.

= Job control system on global scale of CDN/Service delivery network

= RAM (main memory)-aware throughput oriented

= no down time service via load balancing & auto scaling

=> 분산 OS가 DevOps 패러다임을 실현하는 플랫폼이 됨

Why now? (2014)

Ans: 분산 OS에 대한 분명한 이해, CDN, Cloud Computing 성숙

Hadoop architecture와 유사? 누구나 같은 구조를 생각할 수 밖에 없는 구조