

KOREAN STUDENT VERSION

# Beginning Kubernetes



**kubernetes**

Byungwook Hyeon  
&  
Dongwan Kang

# Begining Kubernetes

“Begining Kubernetes” is for kubernetes beginners who want to learn using korean, not a expert. We hope that this book will be a ESSENTIAL for kubernetes beginners.

Byungwook Hyeon & Dongwan Kang

# About the Authors.



**현병욱** Byungwook Hyeon

- 소속 : 한양대학교 ERICA 소프트웨어융합대학 소프트웨어학부
- 이력 :
  - 2007. 03 ~ 2007. 11 : 홍콩 어학연수
  - 2009. 03 ~ 2012. 02 : 창동고등학교 졸업
  - 2014. 03 ~ 2019. 02 : 청주대학교 컴퓨터정보공학과 학사(중퇴)
  - 2016. 02 ~ 2017. 10 : 705 특공연대 특공병, 병장 만기 전역
  - 2019. 03 ~ 2021. 02 : 한양대학교 ERICA 소프트웨어학부 학사
  - 2019. 03 ~ 2019. 06 : (주)대양엔바이오 컨설팅
  - 2019. 09 ~ 2020. 06 : 클라우드 네이티브 컴퓨팅 플랫폼 개발 With KTds
  - 2020. 08 ~ 2020. 12 : 최강학원 수학 강사
  - 2020. 09 ~ 2020. 12 : AWS기반 호텔정보시스템(PMS) 개발, Agora
  - 2020. 10~ 2020. 12 : 대한민국 증권시장에서 거래되는 주식에 대한 R기반 리스크 측도 및 평균 수익률 분석
  - 2020. 11 ~ 2020. 12 : 콜옵션 소유자와 발행자 각각의 거래일에 대한 R기반 현금 흐름과 만기 수익 분석
- 자격 :
  - Microsoft - Azure Fundamental : AZ-900
  - Linux Foundation - Certified Kubernetes Administrator : CKA

# About the Authors.



**강동완** Dongwan Kang

- 소속 : 한양대학교 ERICA 소프트웨어융합대학 소프트웨어학부
- 이력 :
  - 2007. 03 ~ 2010. 02 : 광성고등학교 졸업
  - 2010. 03 ~ 2014. 02 : 유한대학교 컴퓨터소프트웨어공학과 전문학사
  - 2011. 02 ~ 2012. 10 : 3사단 직할 전차대대, 병장 만기 전역
  - 2013. 03 ~ 2017. 01 : 웹플러스 근무, 대리
  - 2019. 03 ~ 2021. 02 : 한양대학교 ERICA 소프트웨어학부 학사
  - 2019. 09 ~ 2020. 06 : 클라우드 네이티브 컴퓨팅 플랫폼 개발 With KTds
  - 2019. 10 ~ 2020. 07 : GS25 안산신안점 근무
  - 2020. 09 ~ 2020. 12 : AWS기반 호텔정보시스템(PMS) 개발, Agora
- 자격 :
  - Microsoft - Azure Fundamental : AZ-900
  - 대한민국 한국산업인력공단 - 정보처리 산업기사

# Table of Contents

---

시작하면서..

---

## Chapter 1. 쿠버네티스 소개

---

Kubernetes란?

Why Kubernetes?

Kubernetes 구성 요소

---

## Chapter 2. 쿠버네티스 컨셉

---

## Chapter 3. 쿠버네티스 APIs

---

## Chapter 4. 쿠버네티스 구성요소

---

## Chapter 5. 보조 프로젝트 소개

---

# 시작하면서..

**저자인 현병욱과 강동완**은 클라우드 엔지니어로의 진출을 준비중인 한양대학교 ERICA 소프트웨어학부 학생입니다. 현병욱과 강동완은 KTds Architecture CoE팀과 클라우드 네이티브 컴퓨팅 플랫폼 개발을 주제로 프로젝트를 수행하여 종합 플랫폼인 Agora를 개발했습니다. 또한, 현병욱은 Linux Foundation과 CNCF사의 kubernetes 운용 능력 검증 시험인 ‘Certified Kubernetes Administrator : CKA’를 취득했습니다.

‘Beginning Kubernetes’는 저자가 프로젝트 수행과 자격 취득 준비 과정에서 경험한 어려웠던 점들을 쉽게 설명하여 입문자들이 쉽게 kubernetes에 접근할 수 있도록 하기위해 집필하게 되었습니다. 이 과정을 통해 kubernetes 오픈 소스 프로젝트가 활성화되길 바랍니다.

**저자 현병욱**

**저자 강동완**

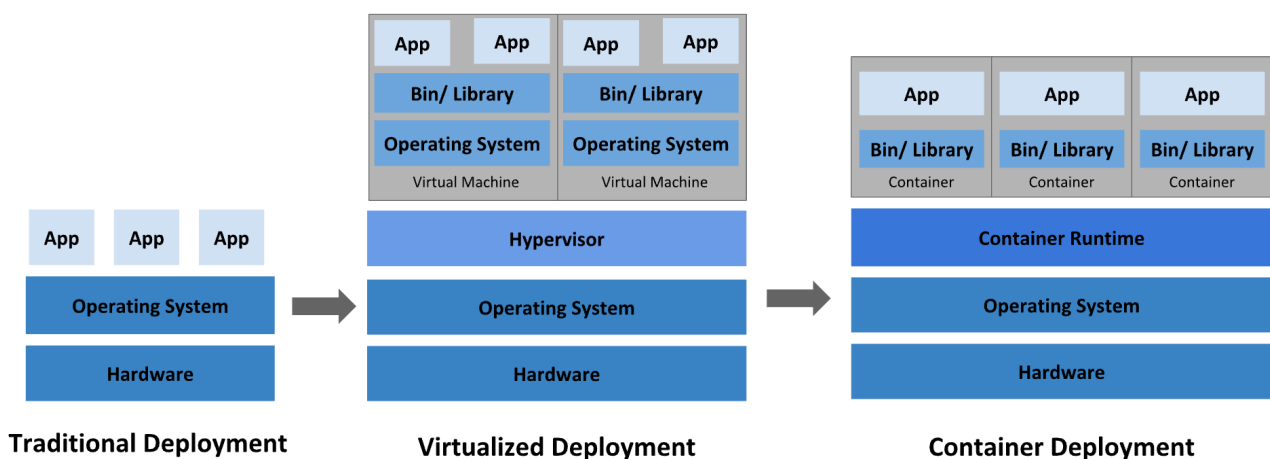
# Chapter 1.

## Kubernetes 소개

### Kubernetes란?

Kubernetes는 컨테이너화된 애플리케이션의 자동 디플로이, 스케일링 등을 제공하는 오픈 소스 기반의 관리 시스템이다. 시스템의 초기 버전은 Google에 의해서 개발되어 2014년 중순에 공개되었다. 프로젝트 당시 kubernetes는 “프로젝트 세븐”으로 시작하였으나, 그리스어로 키잡이(helmsman)를 뜻하는 kubernetes로 2015년 7월 21일에 v1.0이 정식 출시되었다. 현재는 Linux Foundation에서 서비스를 주관하고 있다.

### Why Kubernetes?



초창기 대부분의 애플리케이션은 기본적인 시스템의 물리 서버에서 서비스되었다. 그러나 여러개의 애플리케이션을 구동하고자 한다면 그에 상응하는 수의 물리 서버가 필요하다는 한계가 존재했는데, 이는 서비스 측면에 있어서 애플리케이션의 성능을 저하시킬 수 있다는 치명적인 단점으로 작용했다.

이를 해결하기 위해 VirtualBox, VMware와 같은 가상화 환경이 등장하게 된다. 가상환경은 기존의 단일 물리 서버의 CPU에서 다중으로 실행이 가능하도록하여 애플리케이션의 각 리소스들을 격리하여 성능 저하를 방지할 수 있다. 또한, 격리된 리소스는 서로간의 접근에 있어서 자율성을 제한하여 보안 측면까지 해결이 가능했다. 가상화의 등장은 기존에 요구된 많은 수의 물리 서버의 필요성을 감소시킴으로써 하드웨어 비용의 절감을 효과를 기대할 수 있다. 이는 동적 확장에 자율성을 보장해줄 수 있음은 물론이거니와, 기존 각 물리 서버에서 요구하는 독립적 운영체제의 도입에서 발생하는 비용을 감축킬 수 있기에 다수의 기업들이 가상환경을 도입하기 시작했다.

가상 환경의 도입으로 기존 물리 서버의 시대보다 효율적인 애플리케이션 운용이 가능해졌지만, 가상 환경에도 단점이 존재했다. 바로 리소스간의 접근 제한으로인한 개발 복잡성의 증가이다. 개발 단계에서 가상 환경내의 독립적인 리소스간는 정보를 주고받는 통신에 제약사항을 유발하여 개발 단계를 지연시키는 경우가 발생한 것이다. 이를 해결하기위해 등장하게 된 것이 바로 ‘컨테이너’기술이다. 컨테이너는 VM과 유사하지만 리소스간의 통신 제약사항을 완화하여 애플리케이션 간의 리소스 공유가 가능하도록 구상되었다. 또한, 컨테이너의 가장 큰 장점은 기존의 로우 레벨의 인프라와의 연결 고리가 끊어짐에따라 클라우드나 기타 다른 운영체제로의 이식이 가능해졌다는 점이다.

입문서의 메인 주제인 kubernetes는 컨테이너를 쉽게 관리할 수 있도록 고안된 플랫폼이다. Kubernetes는 분산 시스템에 대한 접근성 강화와 운영 단계를 단축시킬 수 있는 프레임 워크를 제공한다. 이를 통해 Kubernetes는 컨테이너에 대해 애플리케이션의 배포, 확장, 장애 조치, 모니터링, 스케줄링 등을 신속하고 편리하게 처리할 수 있다. 결론적으로 Kubernetes는 컨테이너 시대를 총괄하는 플랫폼인 것이다.

Kubernetes 운용 능력 확보는 4차 산업혁명 흐름에 맞춰 변화중인 애플리케이션 개발 환경에 필수 역량으로 자리잡게 될 것이다.

## Kubernetes 구성 요소

Kubernetes 운용 능력을 향상시키기 이전에 기본적으로 알아야 할 Kubernetes의 구성 요소는 다음과 같다.

### 1. 노드 ( Node )

도커와 같은 패키징 프로그램을 통해 컨테이너화된 애플리케이션을 실행하는 단위이다.

### 2. 파드 ( POD )

파드는 kubernetes에서 생성할 수 있는 가장 작은 단위의 오브젝트로, 애플리케이션의 작은 Instance 이다.

### 3. 클러스터 ( Cluster )

클러스터는 컨테이너화된 애플리케이션을 실행하는 노드의 집합이다.



#### **4. Kube-apiserver**

Kube-apiserver는 kubernetes를 전반적으로 관리하는 메인 server이다. kube-apiserver를 통해 클러스터 안의 모든 동작의 오케스트레이션을 관리할 수 있다.

#### **5. Kubelet**

Kubelet은 kube-apiserver로부터 커맨드를 받아 현재 컨테이너, 노드, 파드 등에 대한 정보를 받거나 노드에 컨테이너를 배포 혹은 삭제하는 기능을 수행한다.

#### **6. Kube-proxy**

Kube-proxy는 서비스와 클러스터 사이의 통신을 보조한다.

#### **7. Kube-scheduler**

Kube-scheduler는 현재 필요한 컨테이너를 가져오는 작업을 수행한다.

#### **8. ETCD**

ETCD는 reliable한 key-value를 간단하게 저장하여 보관할 수 있는 기능을 수행한다. 또한, 현재 클러스터에 존재하는 컨테이너, 파드, 노드 등에 대한 정보를 일괄적으로 백업하여 보관할 수 있다.

#### **9. Replication-controller**

Replication-controller는 노드를 새로운 클러스터에 가져오거나, 사용 불가 상태의 노드를 삭제할 수 있는 작업을 수행한다.

이 이외에도 kubernetes를 구성하고있는 수많은 요소들이 존재하지만, 다음 9가지는 kubernetes를 운용하는데 있어서 가장 기본적으로 숙지하고 있어야 할 구성 요소이다. 자세한 설명은 다음 장에서 진행하도록 한다.

## **Chapter 2.**

# **Kubernetes 컨셉**

























































