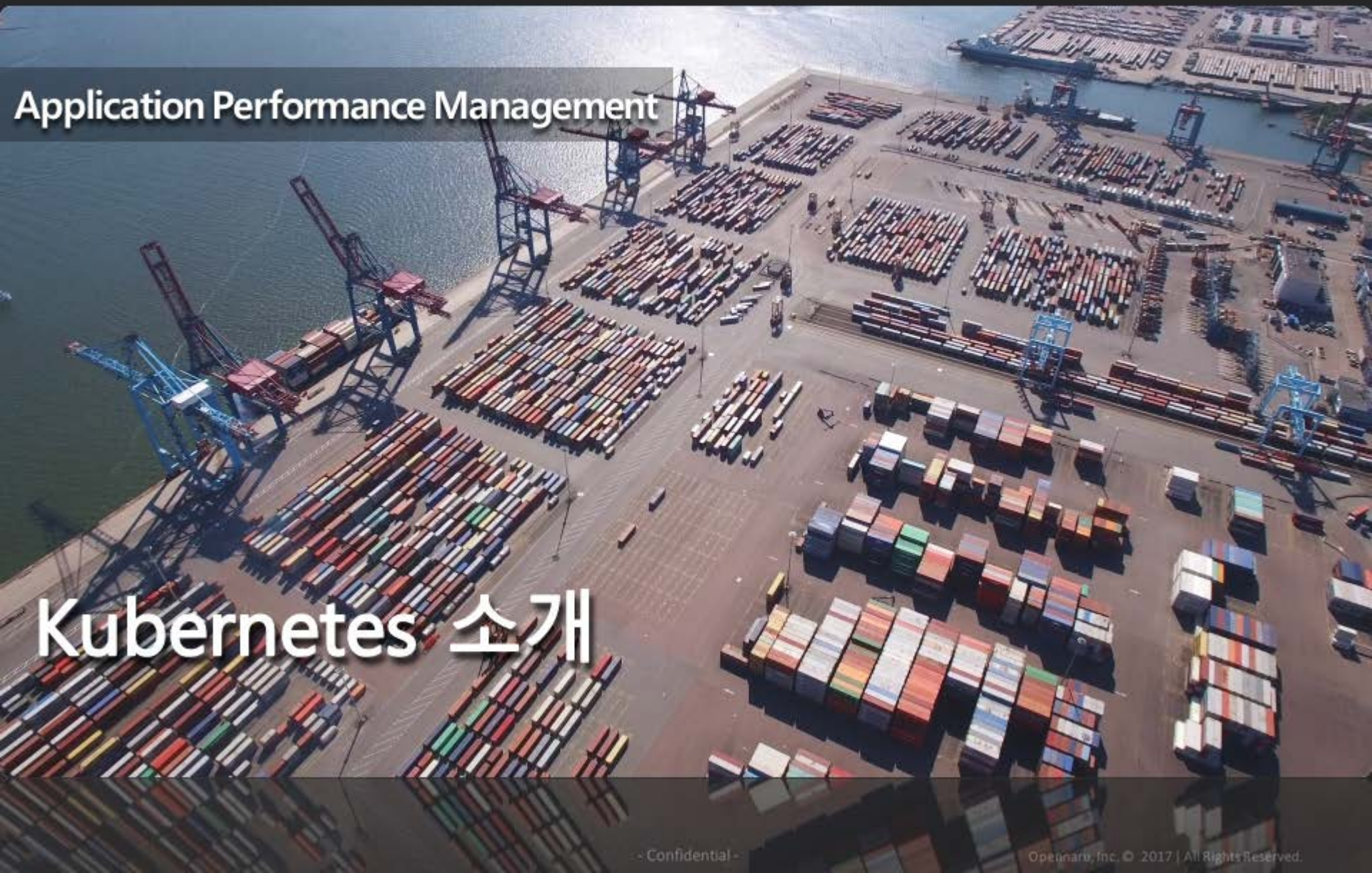

openmaru
APM
컨테이너가 많아질수록
키잡이가 필요합니다.



GOOGLE 과 컨테이너

- **Google의 업무 방식**

Gmail에서 YouTube, 검색에 이르기까지 Google의 모든 제품은 컨테이너에서 실행됩니다.

개발팀은 컨테이너화를 통해 더욱 신속하게 움직이고, 효율적으로 소프트웨어를 배포하며 전례 없는 수준의 확장성을 확보할 수 있게 되었습니다. **Google은 매주 수십억 개가 넘는 컨테이너를 생성**합니다. **지난 10여 년간 프로덕션 환경에서 컨테이너화된 워크로드를 실행하는 방법에 관해 많은 경험**을 쌓으면서 Google은 커뮤니티에 계속 이 지식을 공유해 왔습니다.

초창기에 cgroup 기능을 Linux 커널에 제공한 것부터 **내부 도구의 설계 소스를 Kubernetes 프로젝트로 공개**한 것까지 공유의 사례는 다양합니다. 그리고 이 전문 지식을 Google Cloud Platform으로 구현하여 개발자와 크고 작은 규모의 회사가 최신의 컨테이너 혁신 기술을 쉽게 활용할 수 있도록 하였습니다.

Source - https://cloud.google.com/containers/

About Kubernetes

openmaru
APM

쿠버네티스(K8s)는 컨테이너화된 애플리케이션을 자동으로 배포, 스케일링 및 관리해주는 오픈소스 시스템

- 그리스어로 키잡이의 어원

  " Helmsman (키잡이) " 그리스어.

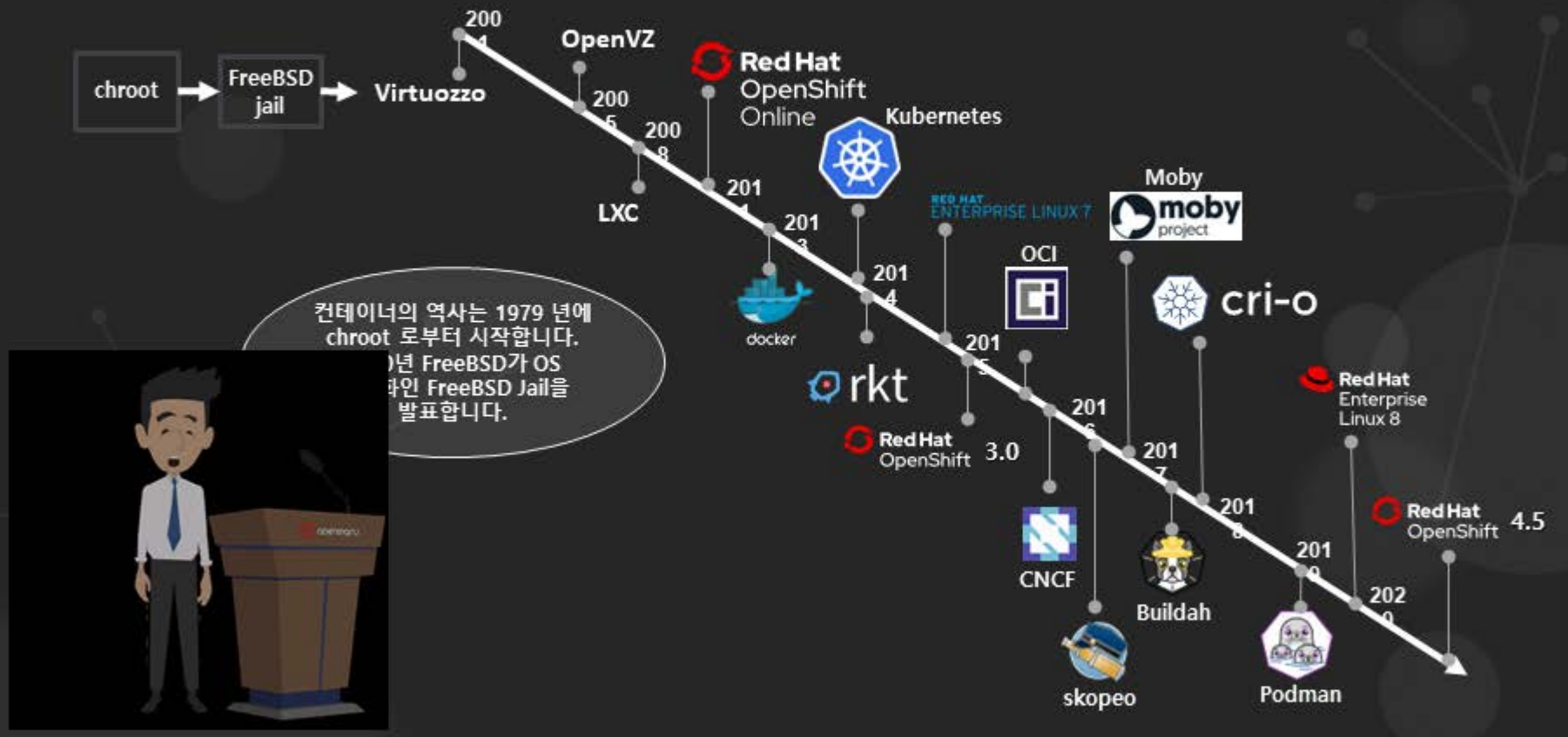
  " Governor (지사) " 의 어원

- k8s (ubernete-> 8) called k-eight-s
- Originally designed by Google

  (Borg-> Omega-> Kubernetes)
- Go로 작성된 오픈 소스 ,

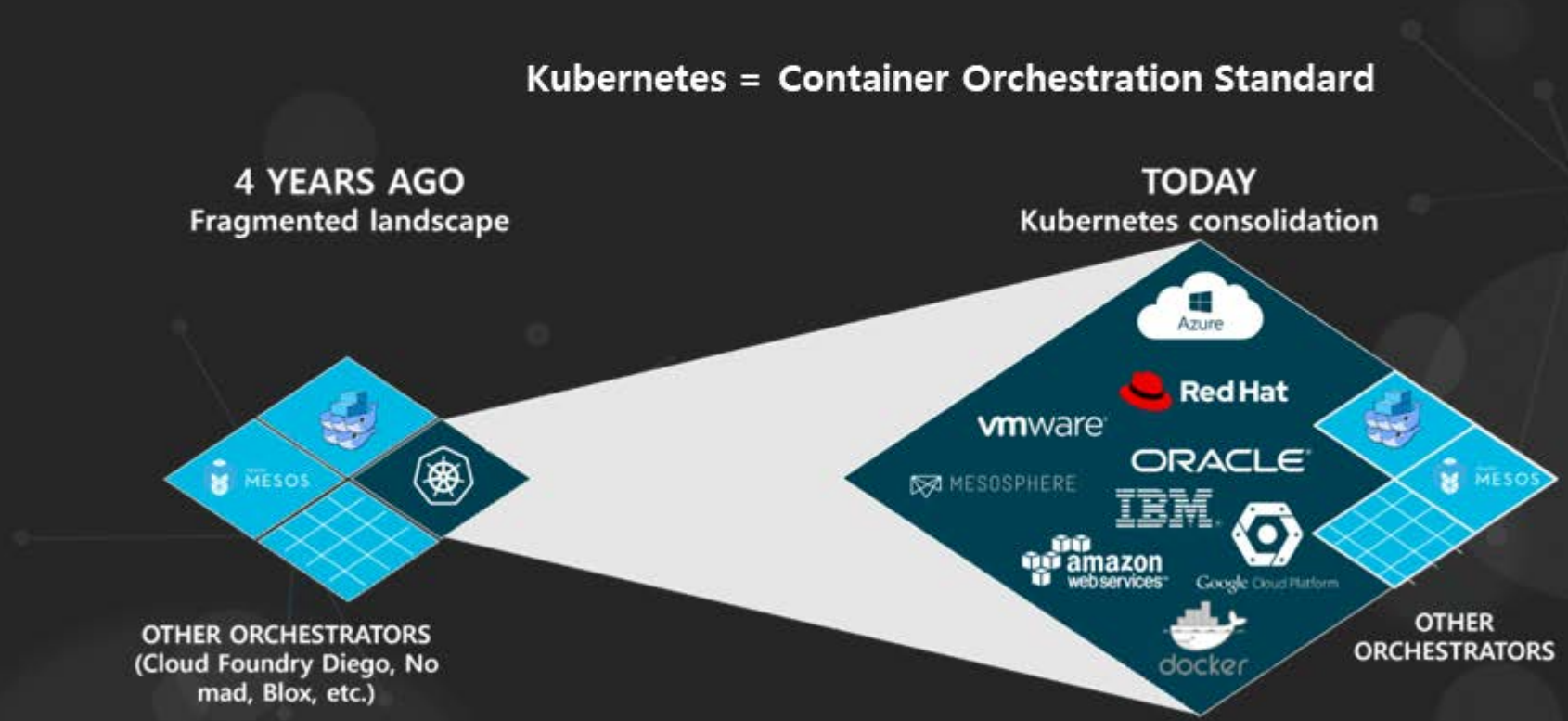
  OSS (Apache License 2.0)

Container와 Kubernetes의 역사



Kubernetes는 컨테이너 오케스트레이션의 표준

openmaru APM



Kubernetes는 누구에게 필요한 도구 일까요?

- IT 운영 부담을 줄이려고 하는 개발팀,

  시스템 운영팀, 네트워크 운영팀, DEVOPS 팀

  - 애플리케이션 변경 요건이 많고,하루에 여러 번 배포
  - 이벤트로 인한 부하에 대응할 수 있는 시스템 요구
  - 빈번한 설정 변경과 시스템 작업으로 인한 이력 관리
  - 복잡한 작업 절차서
  - 배포 정책 요구와 롤백 방안

Containers and Kubernetes: The Time Is Now

openmaru APM

CONTAINERS IN **DEVELOPMENT**

CONTAINERS IN **PRODUCTION**

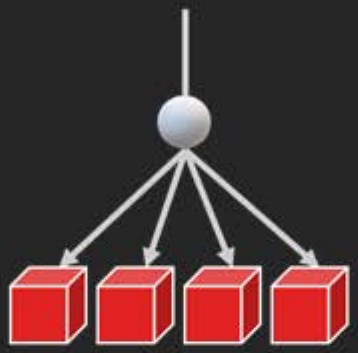
Source : https://blogs.vmware.com/cloudnative/2018/01/23/containers-kubernetes-benefits/

Kubernetes 주요 기능

openmaru APM

Scale Out / In

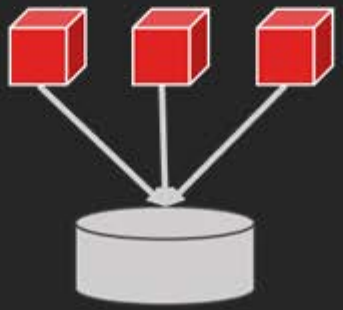
Load Balancer



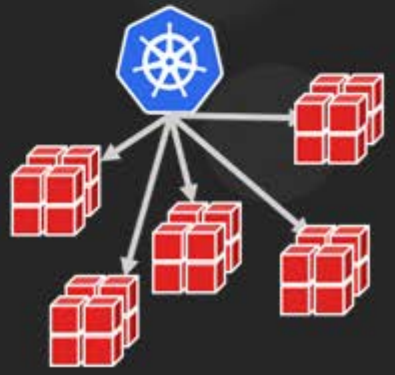
Rolling Update

Auto Healing

Persistence Volume



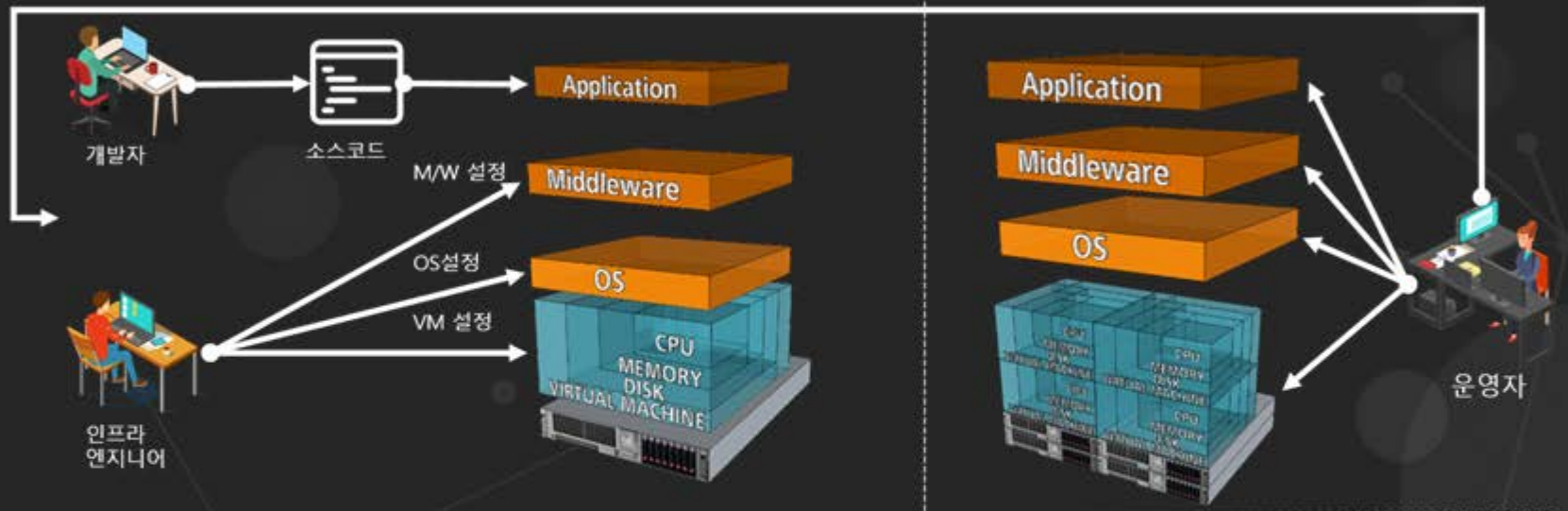
Container Orchestration



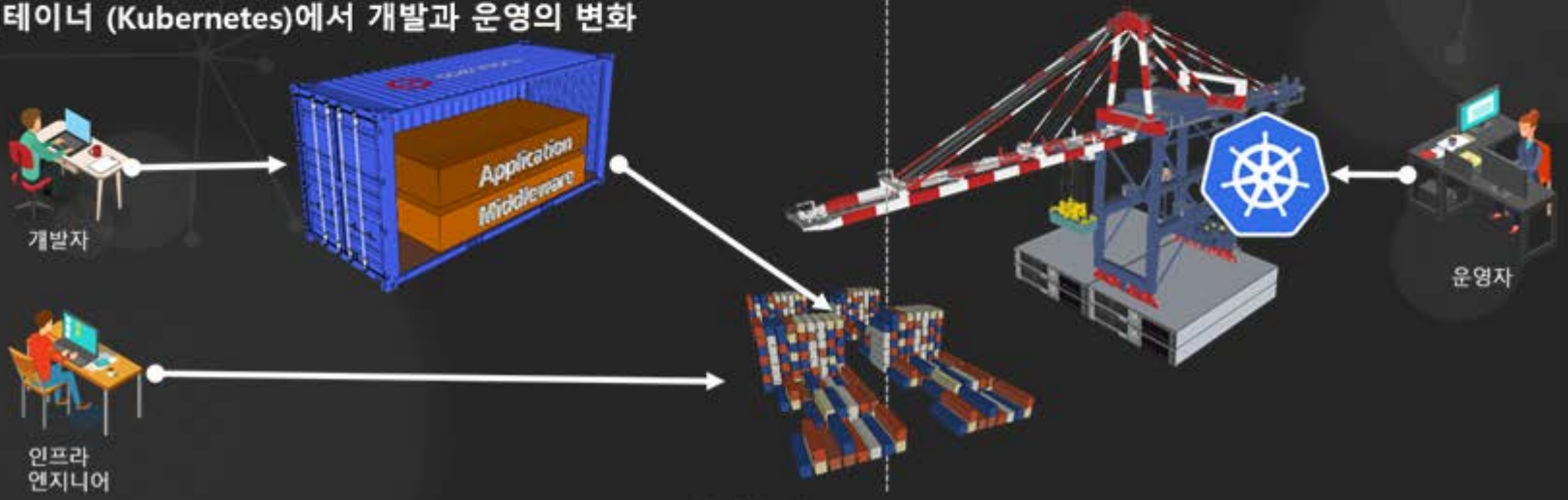
개발과 운영의 변화

openmaru APM

- 가상화 (Virtualization) 에서 개발 및 운영



- 컨테이너 (Kubernetes)에서 개발과 운영의 변화



Application Performance Management

Kubernetes Architecture

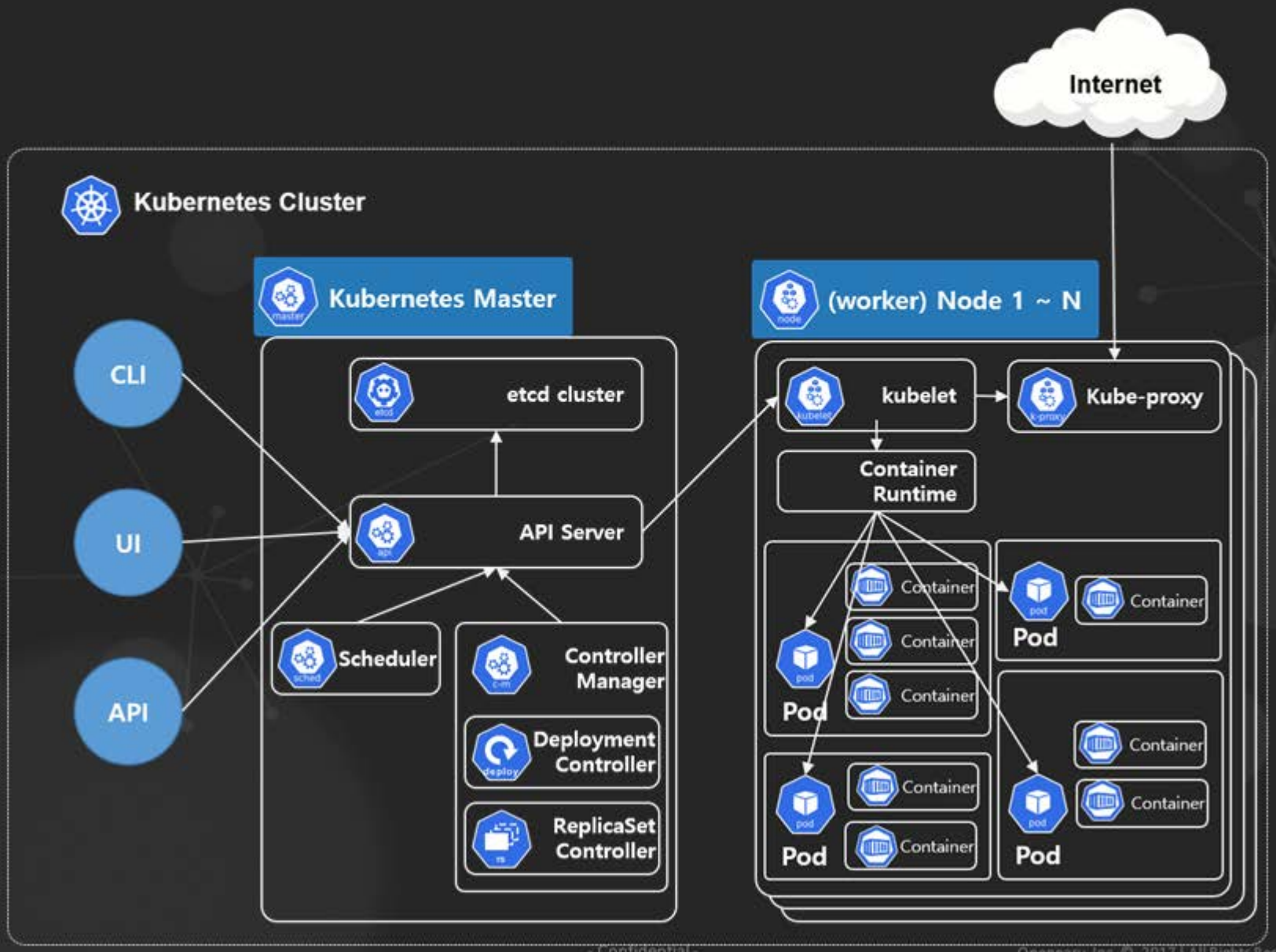
Kubernetes Architecture

openmaru APM

Kubernetes는 역할에 따라 2가지의 Node로 구분 됩니다.

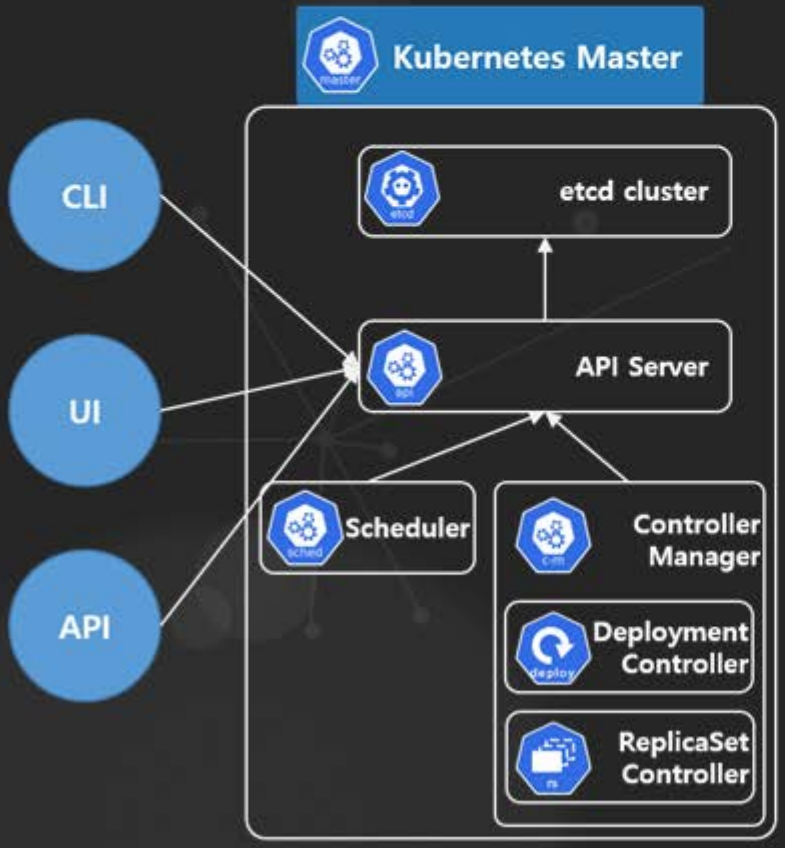
1. Cluster를 관리하는 [ ] 노드
2. 서비스 애플리케이션이 동작하는 [ ] 노드

Kubernetes Architecture



Kubernetes Master

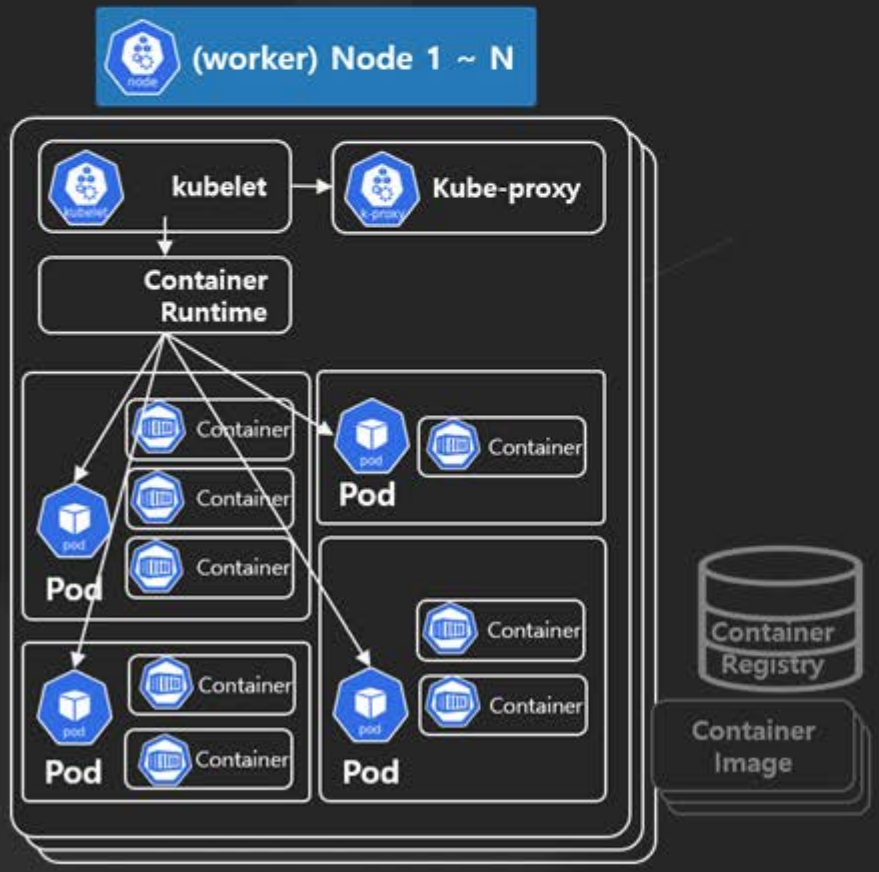
- 마스터는 컨테이너를 관리하는 역할로 "컨트롤 플레인" 이라고 불림
- Master는 각 Node 에 배포, 업데이트, 스케일링 등의 지시를 내리는 역할
- 3개 이상의 서버에서 마스터를 구성 할 것을 권장



컴포넌트	설명
API Server	• 쿠버네티스 클러스터의 모든 작업을 제어하는 RESTful API • 쿠버네티스 컨트롤 플레인에 대한 프론트엔드
Scheduler	• 노드가 배정되지 않은 새로 생성된 Pod를 감지하고 그것이 어느 Node에서 실행될 지를 선택 • Pod 를 어떻게 Node 에 배치할지를 예약
Controller-Manager	• 백그라운드에서 클러스터 컨트롤러를 관리 . API 서버를 이용하여 클러스터 상태 모니터링 • 클러스터를 설계한 상태로 유지
etcd	• 클러스터의 모든 정보를 저장하는 데이터 저장소 • 고가용성을 가진 Key-Value 저장소 구조

Kubernetes (worker) Node

- 노드는 마스터의 지시로 컨테이너를 실행하는 역할을 하며 "데이터 플레인" 이라 불림
- 노드는 최소 1 개로 구성 할 수 있지만 Kubernetes는 두 대 이상의 노드를 구성 할 것을 권장



컴포넌트	설명
kubelet	• 노드에서 실행되는 에이전트로 컨테이너가 Pod에서 실행 중인지 확인 • kubelet은 다양한 메커니즘을 통해 제공되는 PodSpec을 가져와 해당 요구사항으로 상태를 유지
pod	• container 의 그룹 k8s 의 최저 배포 단위 • Pod 중 컨테이너 Storage, namespaces , port 공유
kube-proxy	• kube-proxy는 호스트 상에서 네트워크 규칙을 유지하고 연결에 대한 포워딩을 수행함 • 쿠버네티스 서비스를 추상화
Container runtime	• 작성된 이미지를 가져와 컨테이너를 실행 • Kubernetes 는 컨테이너 런타임 교체가 가능하기 때문에, Docker , containerd , rkt , cri-o 같은 컨테이너 런타임을 사용

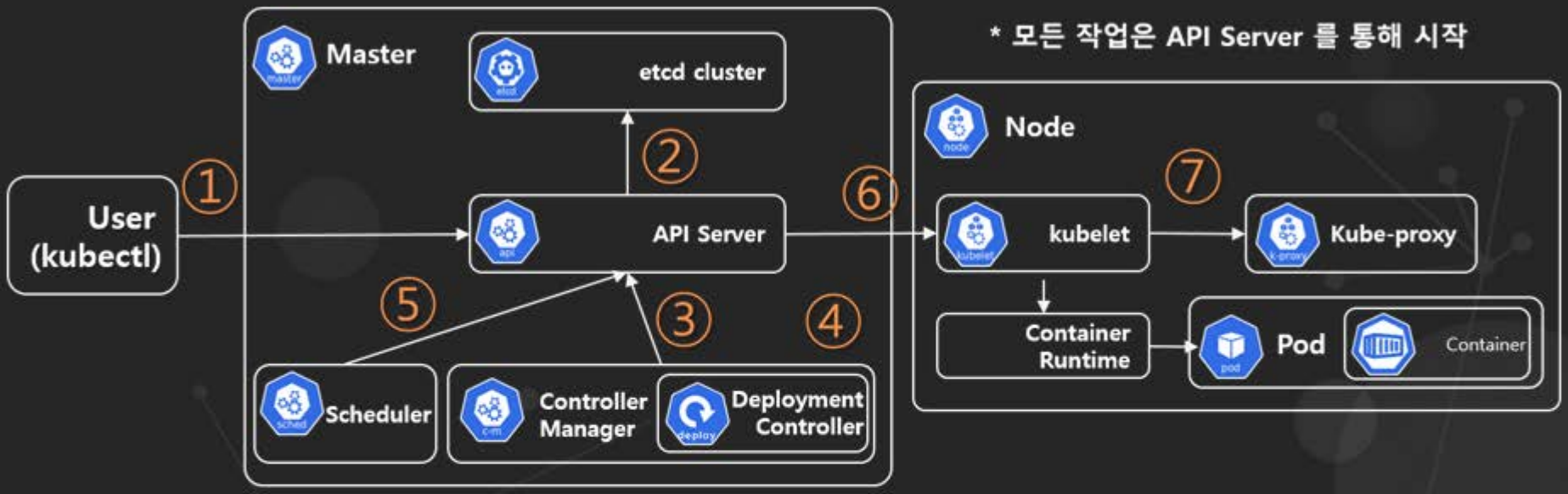
Kubernetes Architecture

Kubernetes 운영자 혹은 개발자는 명령어를 입력해야 합니다.

Kubernetes API Server가 그 명령어를 입력받고 수행하는데요.

1. 명령어를 수행할 수 있는 API Server가 기동중인 노드는 [ ] 노드

Kubernetes Architecture



① 사용자는 "kubectl"를 사용하여 컨테이너 배치에 필요한 정보 (컨테이너 이미지 개수 등)를 전달

② API Server는 받은 내용을 etcd cluster (DB)에 저장

③ Controller는 자원의 변화를 감지하면 원하는 상태가 되도록 동작

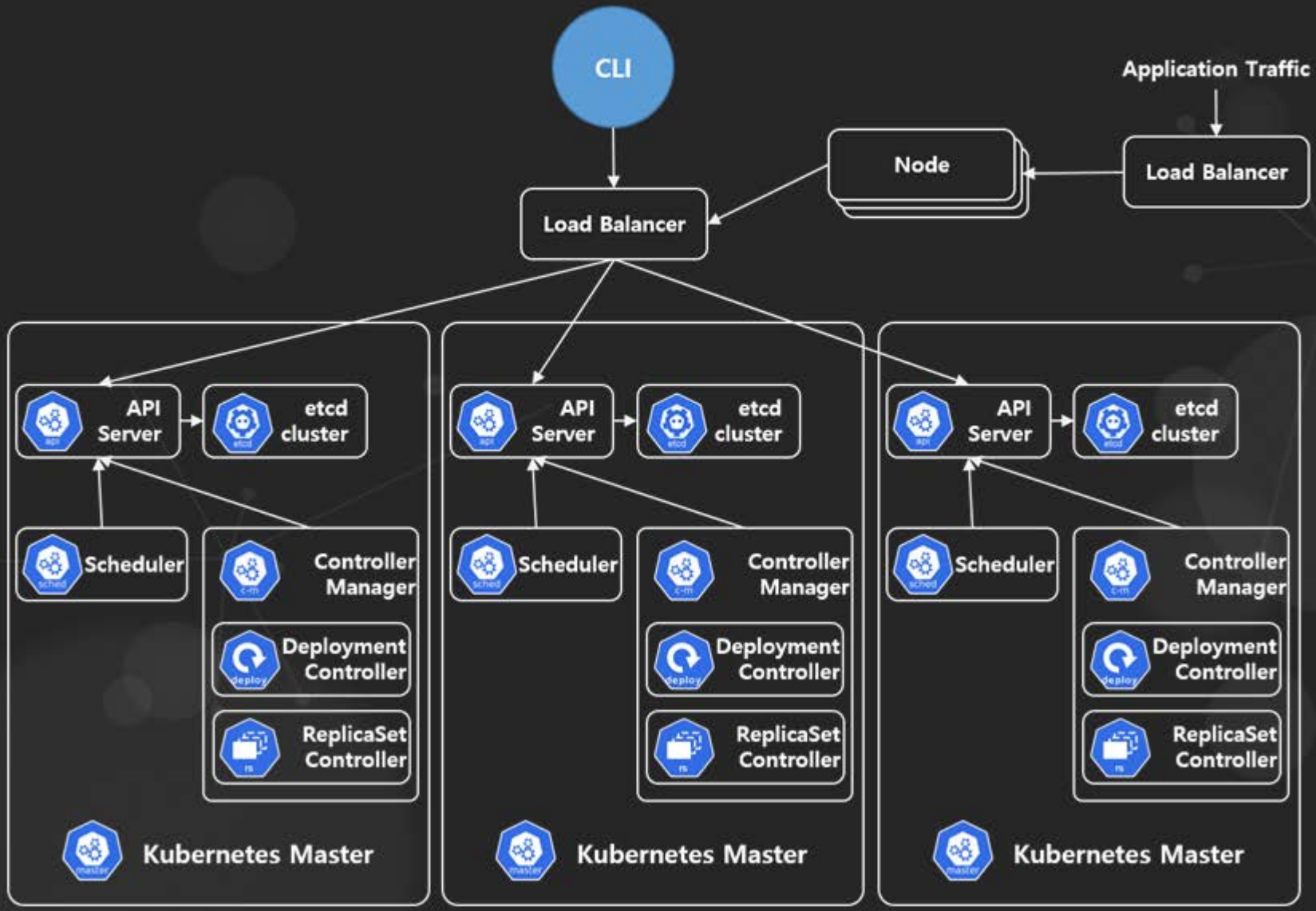
④ Deployment Controller는 새로운 Pod 정보를 API Server를 통해 DB (etcd)에 저장

⑤ Scheduler는 정보에 따라 배포 대상 Node를 결정

⑥ Kublet 는 Container Runtime 를 사용하여 Pod를 생성

⑦ Kube-Proxy는 클러스터의 내부 또는 외부에서 Pod에 대한 접속 방법 제공

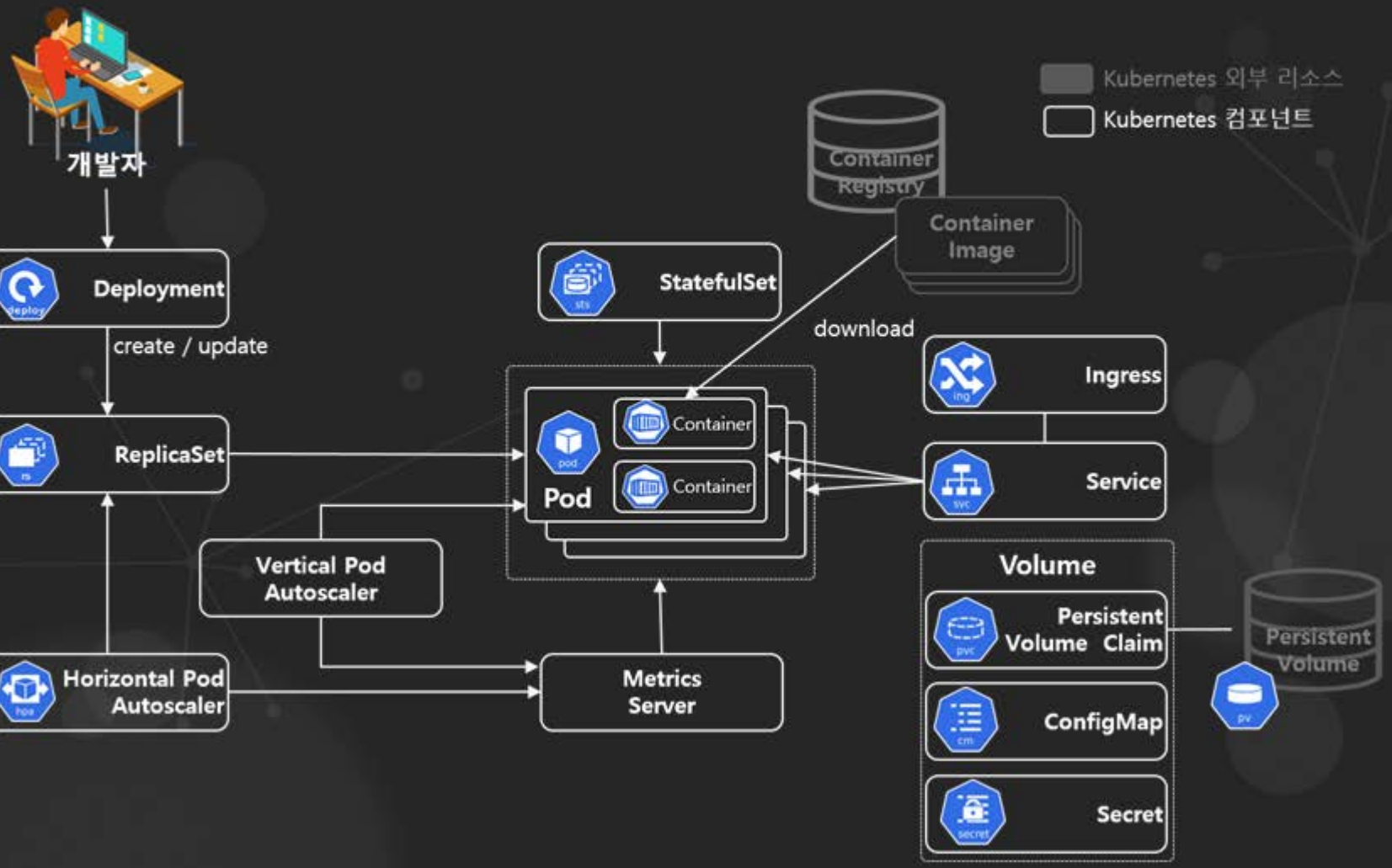
Kubernetes Master 고가용 구성



Kubernetes Architecture

Kubernetes에서 컨테이너들을 기동시키고 관리하는 최소 단위를 [ ] 라고 합니다.

Kubernetes – POD 운영 프로세스



Kubernetes Architecture

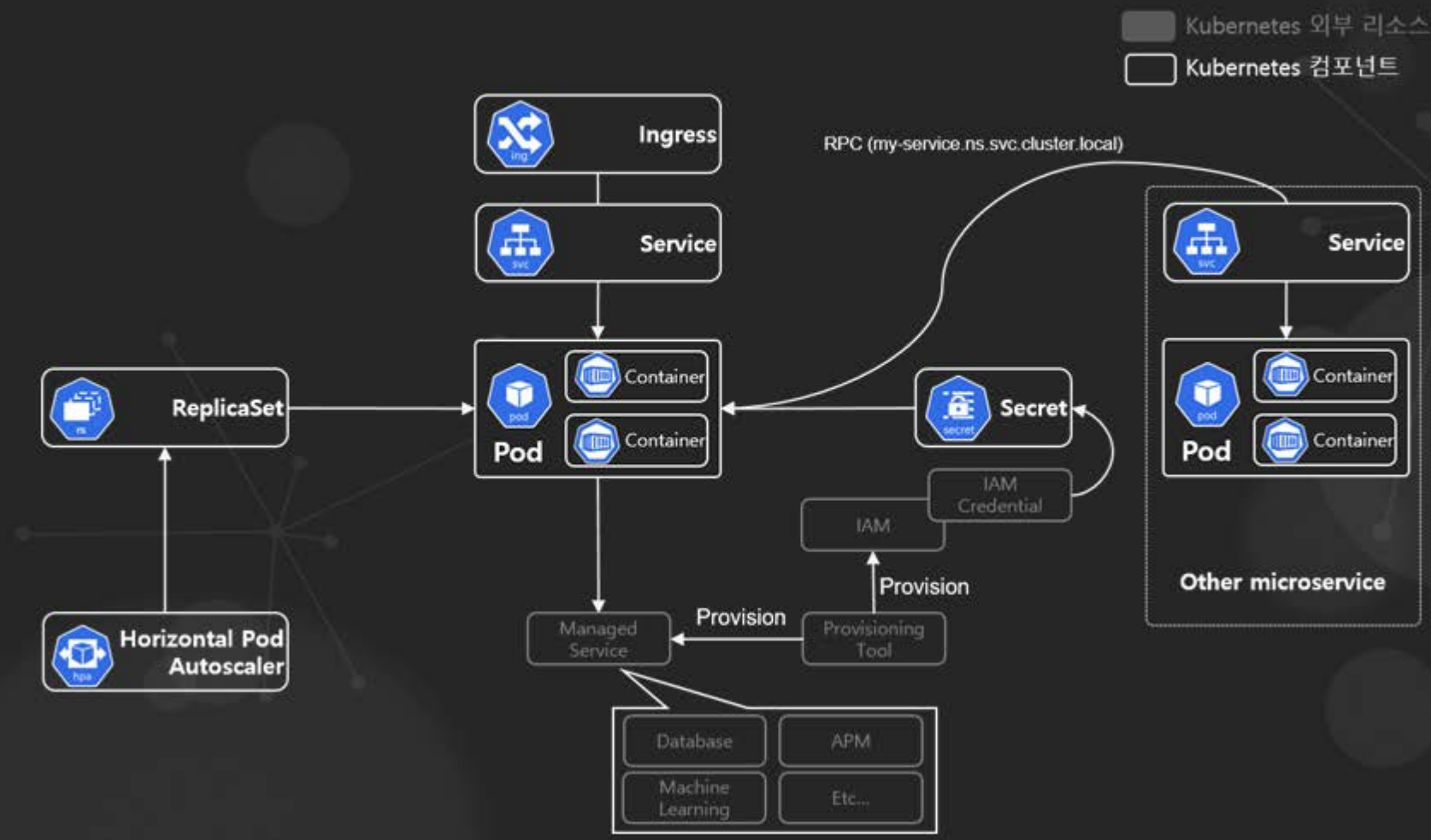
openmaru APM

Kubernetes는 SDN을 통해 Pod 네트워크와 Service 네트워크가 생성되고 Kubernetes 내부 통신은 이 네트워크로 통신합니다.

1. 일반적으로 Kubernetes 내부의 Pod들은 [ ] 를 이용하여 다른 Pod끼리 통신합니다.

Kubernetes – 서비스 호출 프로세스

openmaru APM



Kubernetes Architecture

openmaru APM

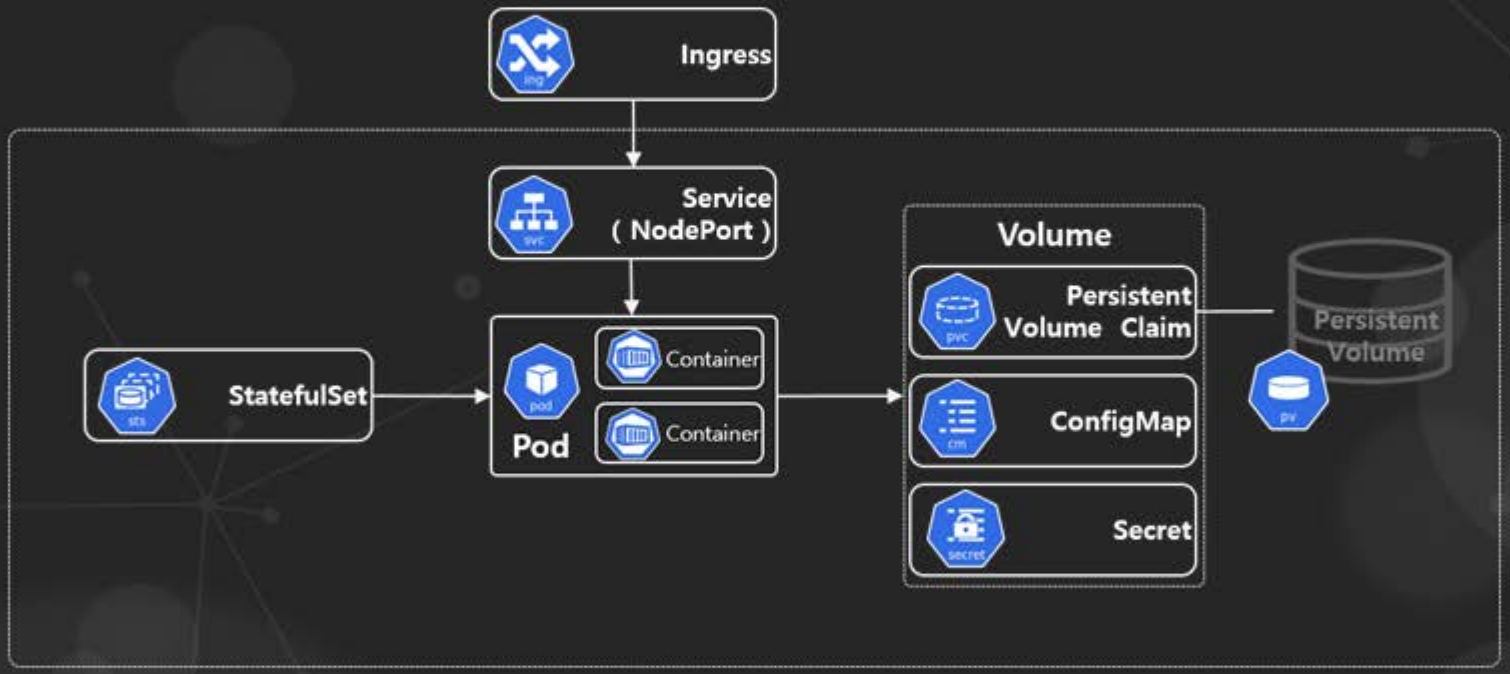
컨테이너는 휘발성이라는 특성을 가지고 있습니다.

그렇기 때문에 저장해야하는 데이터는 별도로 설정을 해야하는데요.

1. [ ] Volume과
   [ ] Volume Claim이 필요합니다.

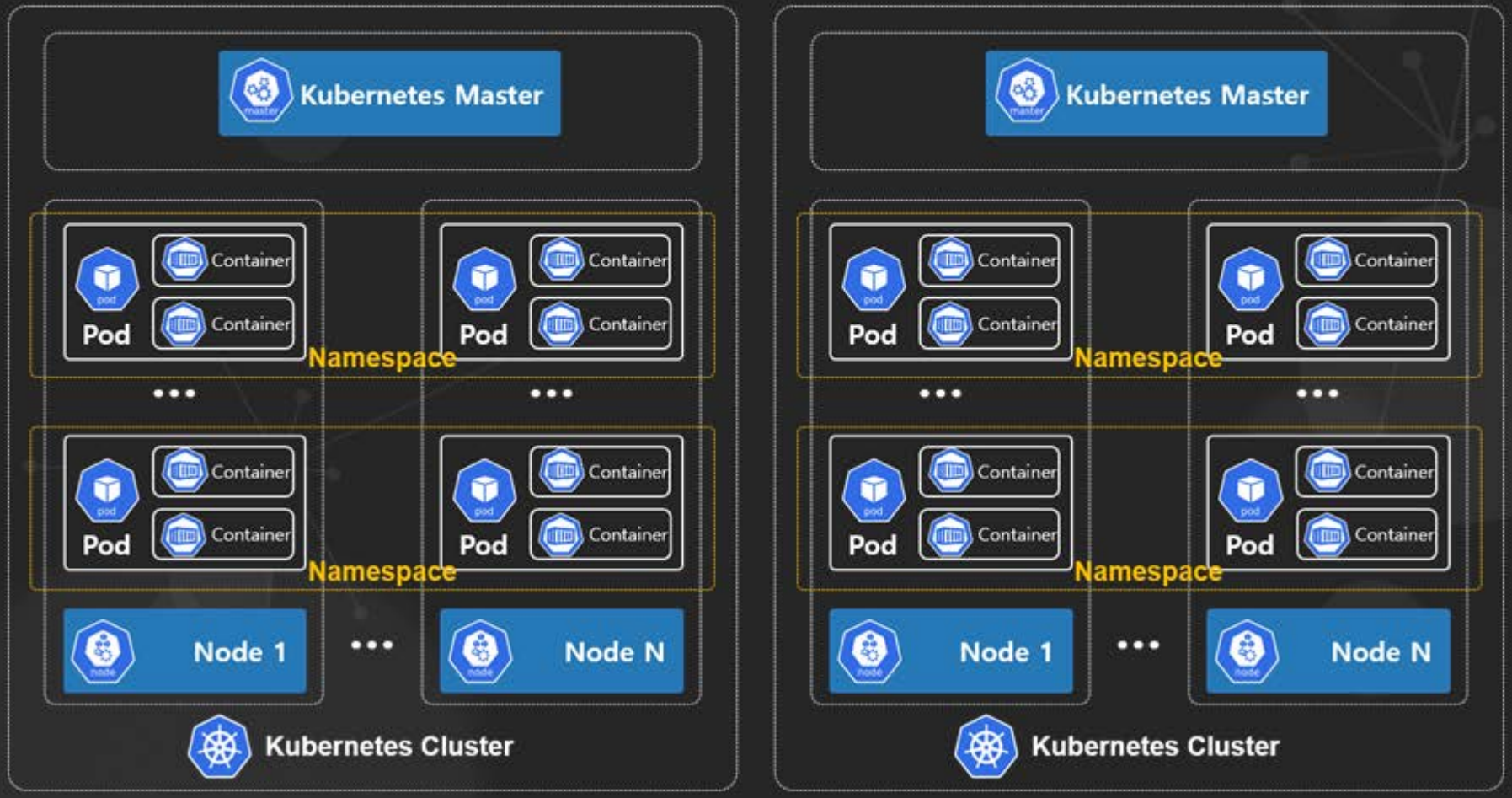
Kubernetes – Application 구성 단위

openmaru APM



Kubernetes – Master/Node

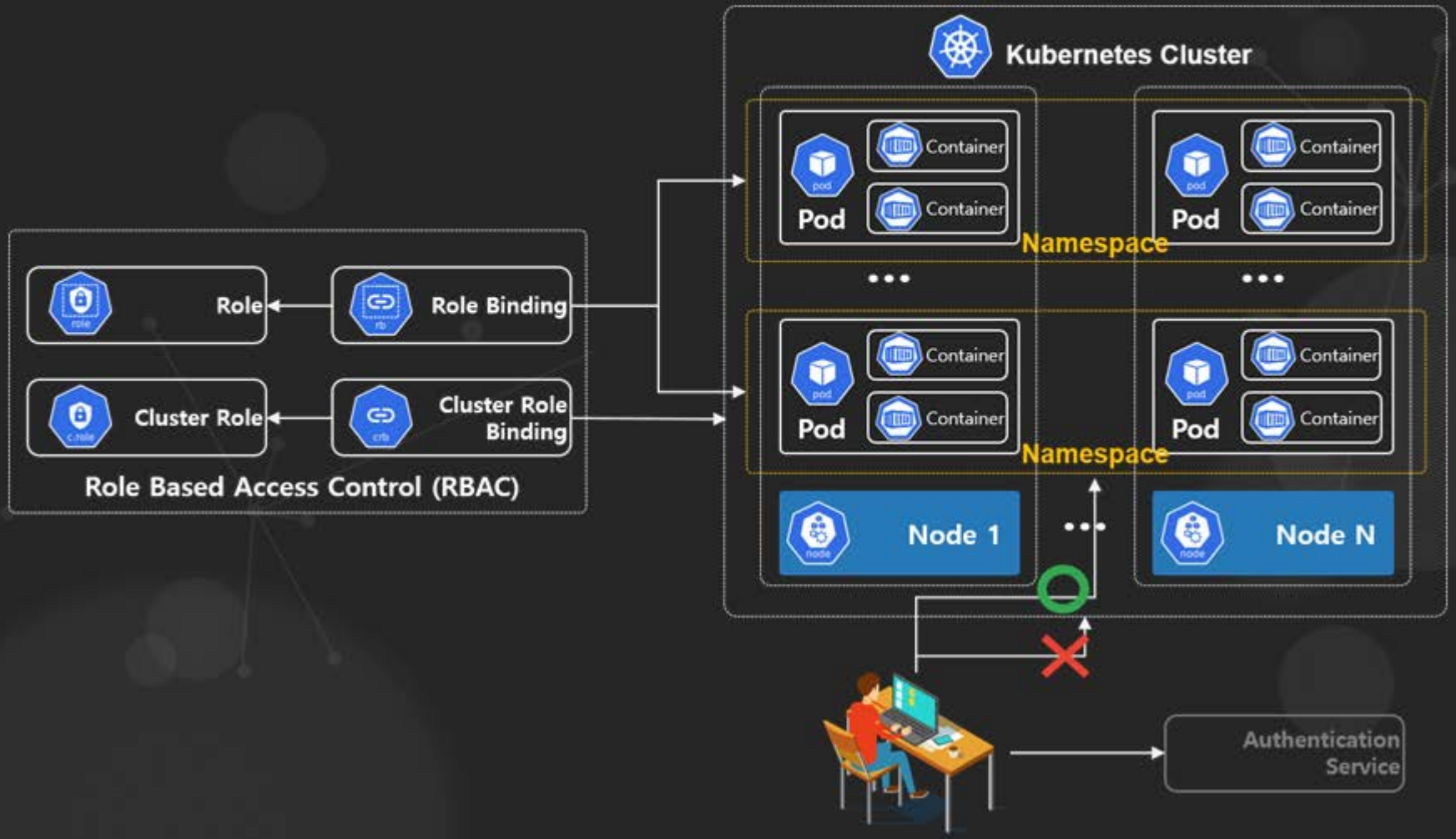
- Kubernetes Cluster 단위



Kubernetes – 권한 관리

openmaru APM

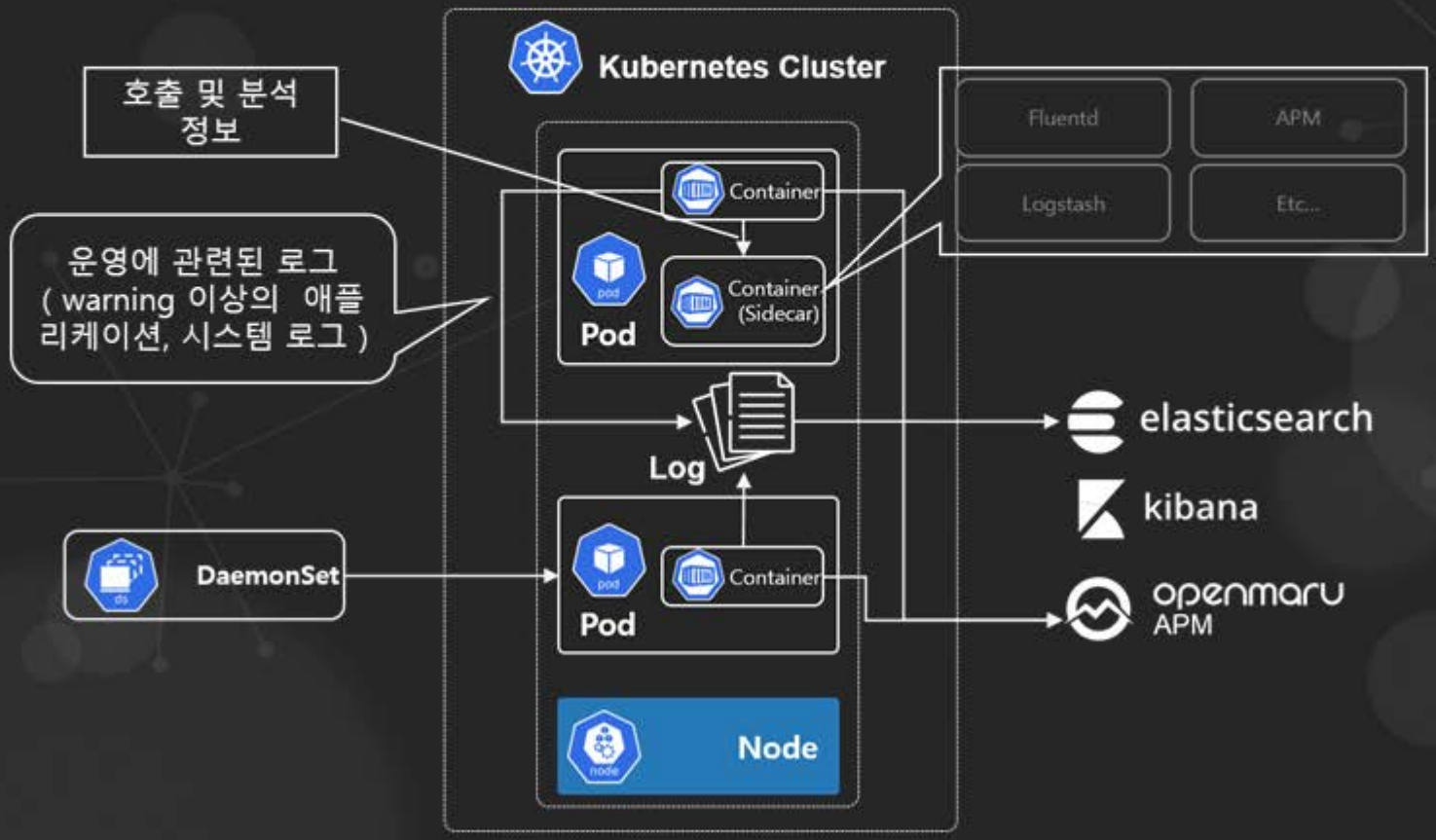
- Role Based Access Control (RBAC)



Kubernetes – Log 통합과 모니터링

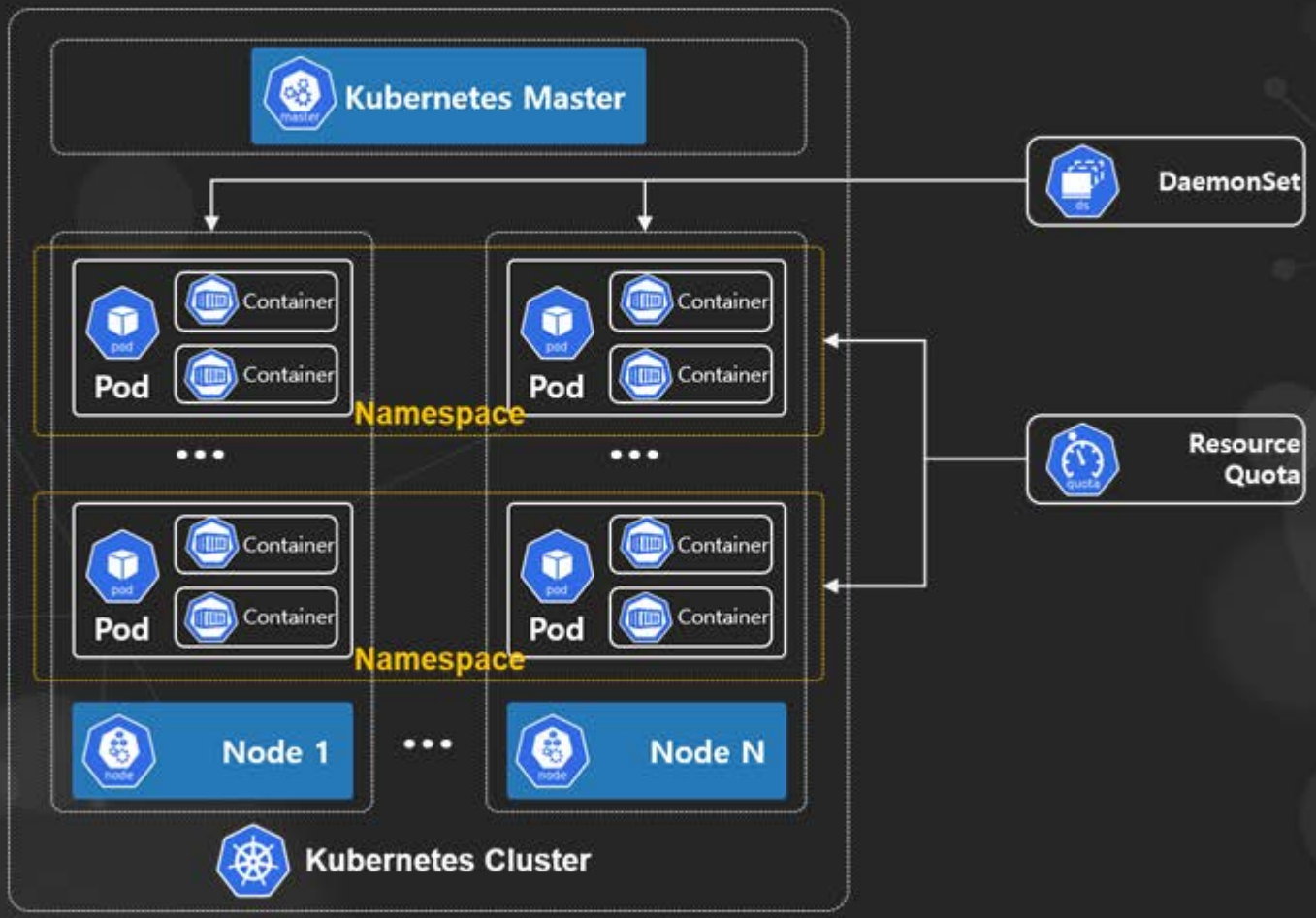
openmaru APM

- 간단한 설정을 통한 로그 통합 및 모니터링
- 과대한 로그 전송으로 인한 시스템 장애를 방지하고 안정적인 운영 기반 제공



Kubernetes - Quota

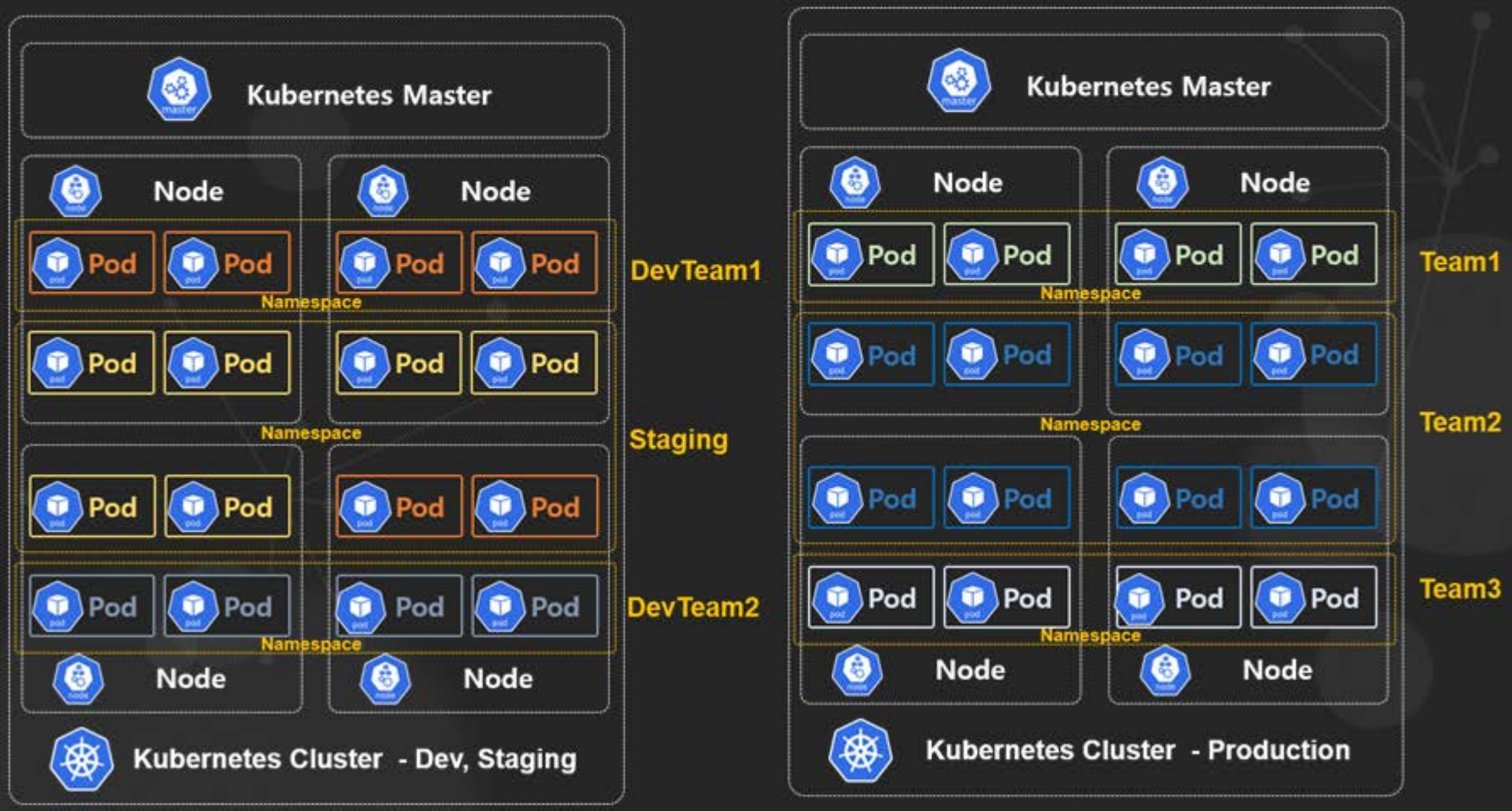
openmaru APM



Kubernetes – 업무별 구분

openmaru APM

- Kubernetes Cluster 단위



제품 / 서비스에 관한 문의

- 콜 센터 : 02-469-5426 (휴대폰 : 010-2243-3394)
- 전자 메일 : sales@opennaru.com