

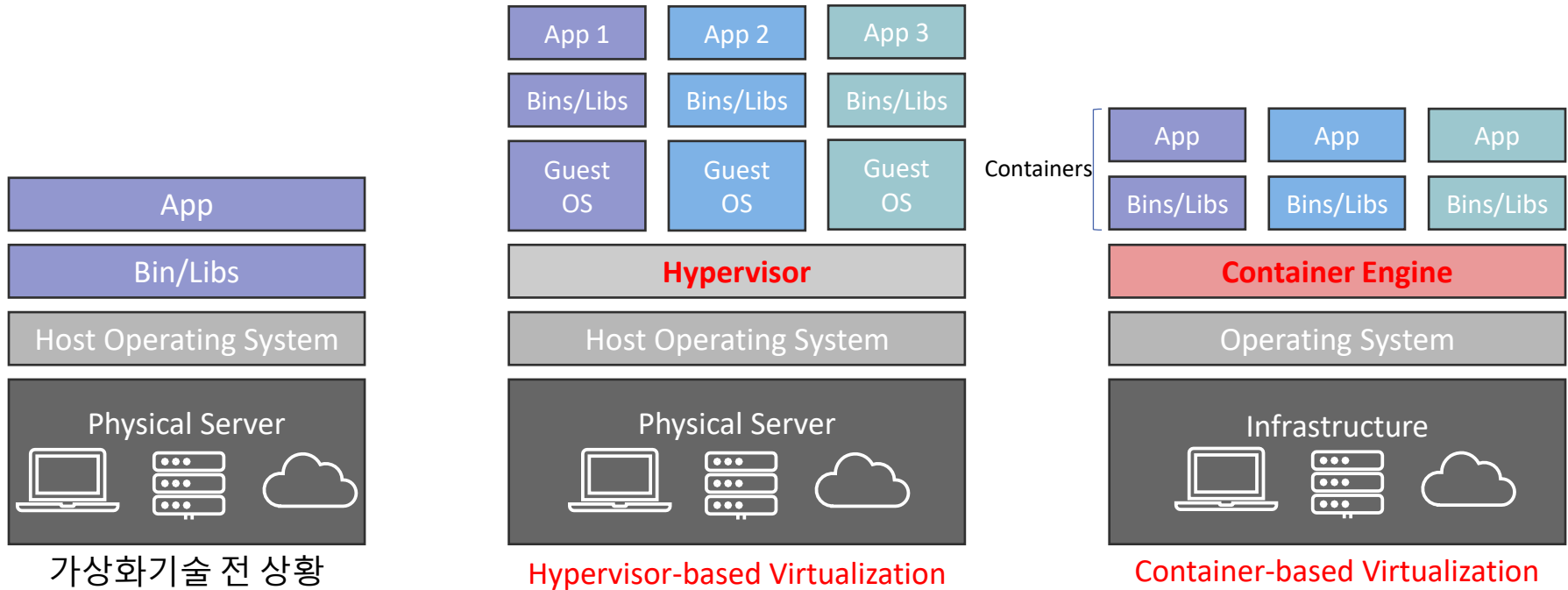
Kubernetes 기초

과학기술정보통신부 스타랩

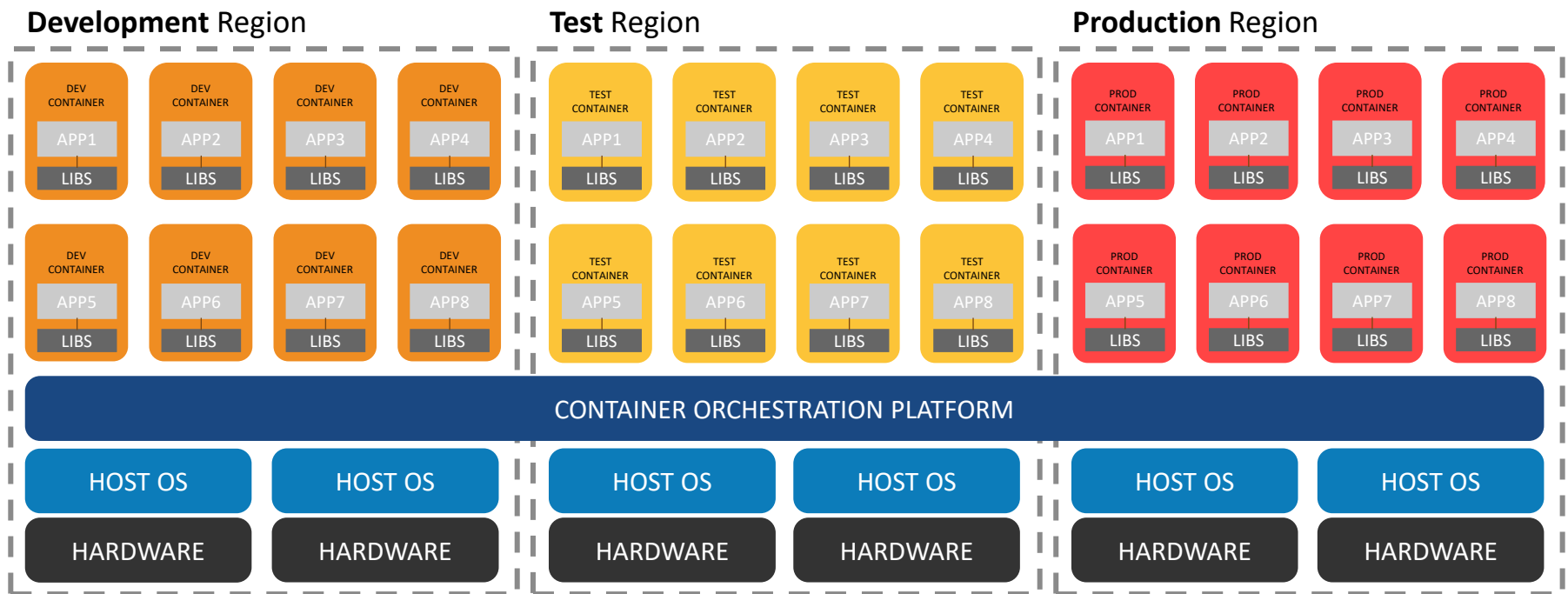
Distributed Cloud and Network Research Lab

김영한

EVOLUTION OF VIRTUALIZATION



CONTAINER ORCHESTRATION PLATFORM



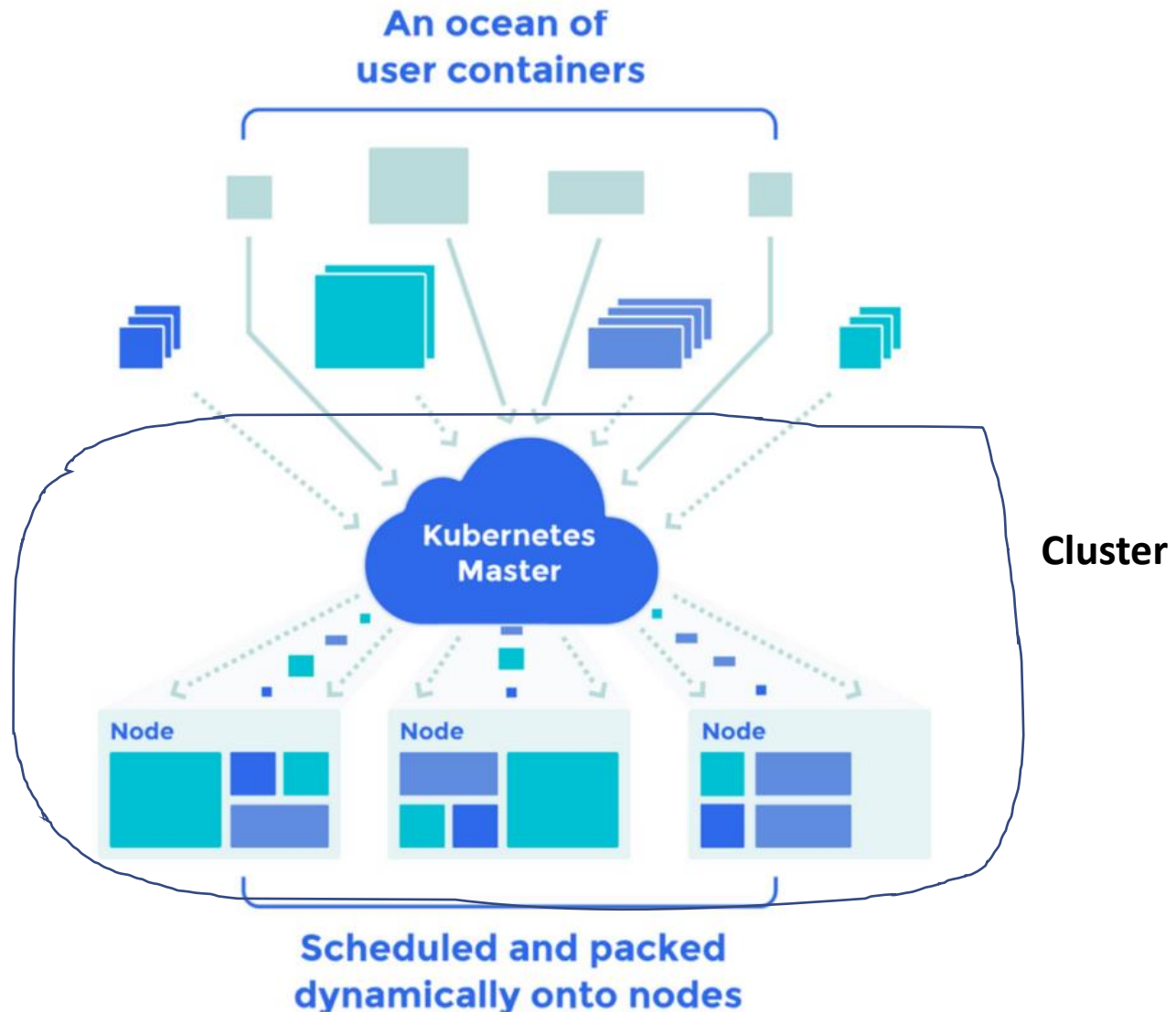
CONTAINER Engine, ORCHESTRATION

	Container Engine	Container Orchestration
Function	Keep software separated into its own “clean” view of an operating system	Management of container deployment, scaling, connect
Predecessors/ Alternatives	<ul style="list-style-type: none">- Virtual Machines- Direct installation	<ul style="list-style-type: none">- Homegrown scripts- Manual, static configuration between containers
Packages/Vendors	<ul style="list-style-type: none">- Docker- RKT- LXC- Mesos- CRI-O	<ul style="list-style-type: none">- Kubernetes- Docker swarm- Mesos

KUBERNETES

- ✦ Born from a Google internal project in mid-2014 (Google “Borg”)
- ✦ 1.0 Release in July 2015
- ✦ Google partnered with the **Linux Foundation** to form the **Cloud Native Computing Foundation (CNCF)** to offer Kubernetes as an open standard
- ✦ Frequently abbreviated “**k8s**” – Greek for “helmsman” or “**pilot**”

Kubernetes Cluster: Master + (worker) Nodes



KUBERNETES working on any ...

✦ Kubernetes **runs anywhere**

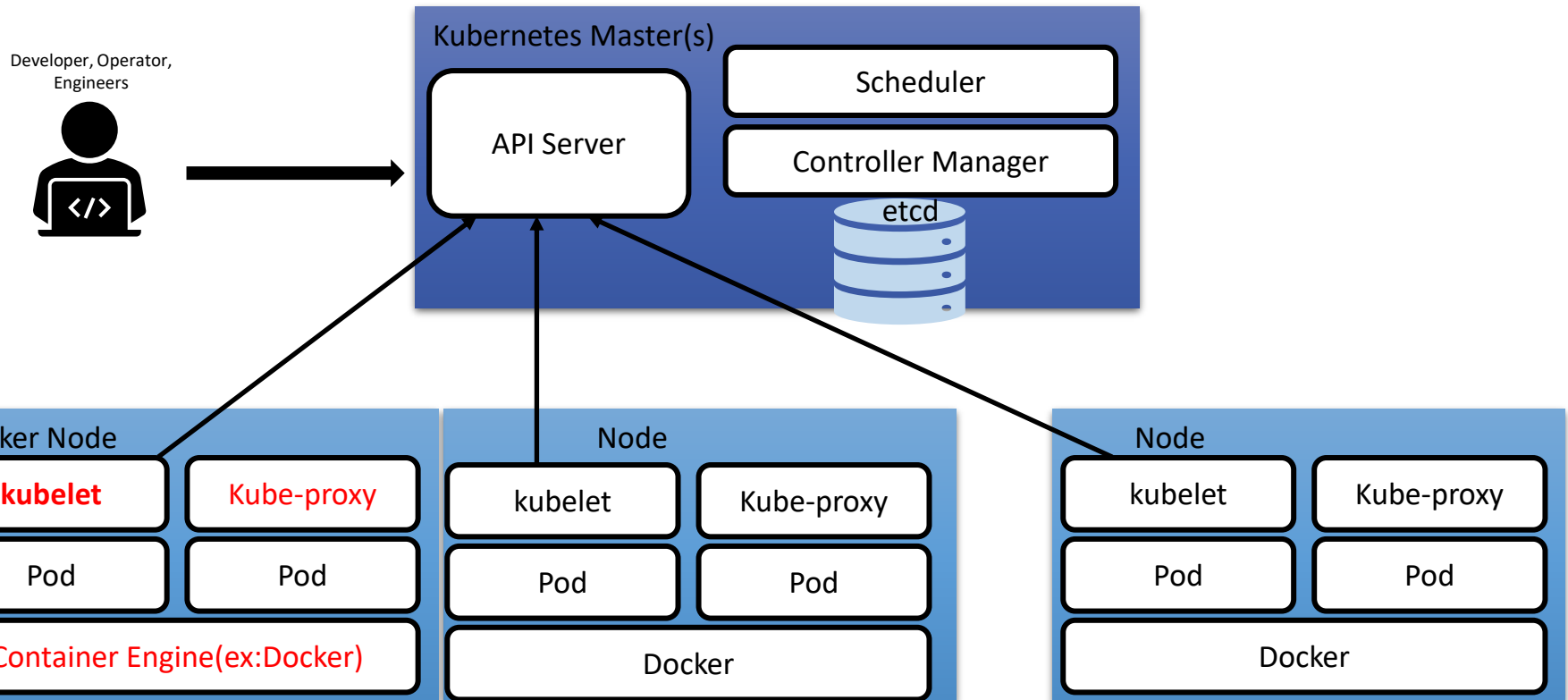
- ✦ Your laptop
- ✦ Globally distributed data centers
- ✦ Major cloud providers
- ✦ Anywhere in any combination

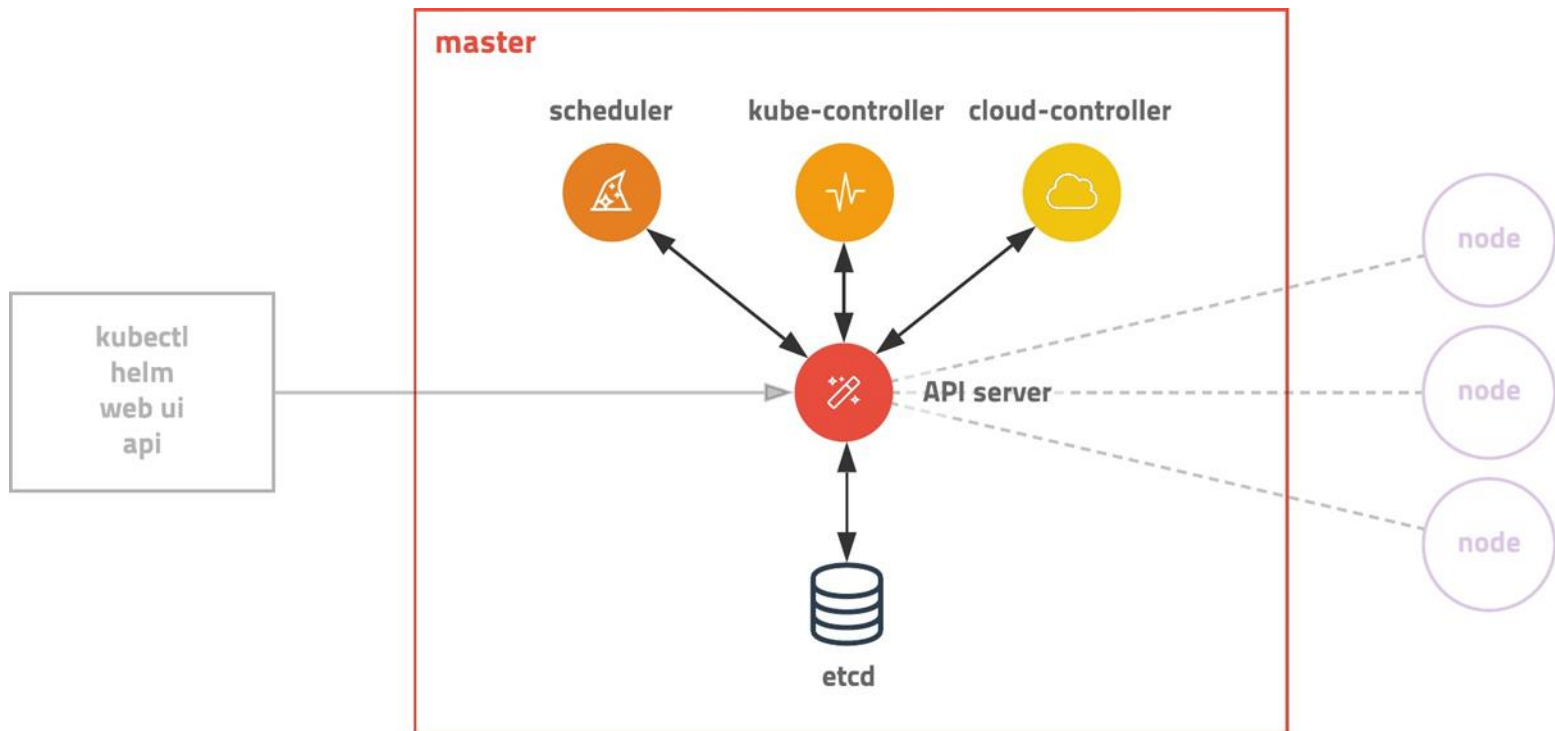
ARCHITETURE OVERVIEW

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

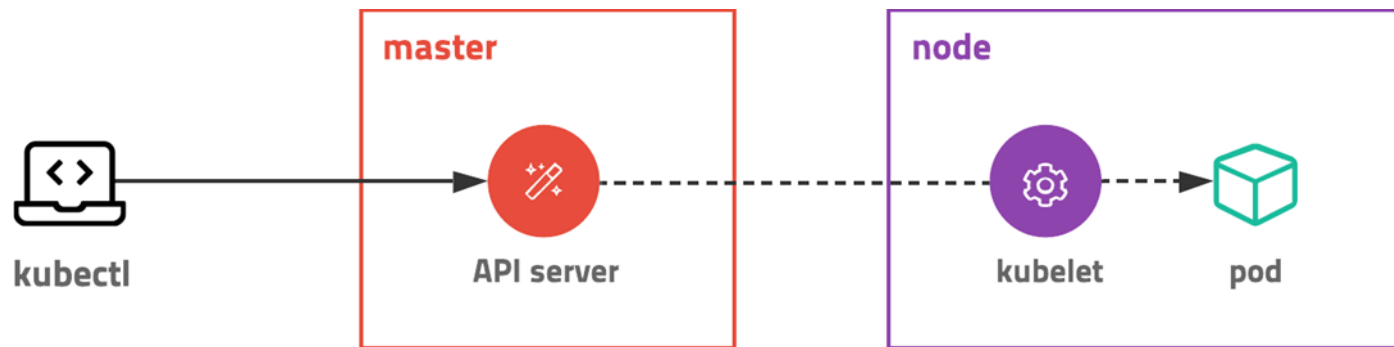
쿠버네티스는 컨테이너화된 응용 프로그램의 배포, 확장 및 관리를 자동화하는 오픈 소스 시스템입니다

KUBERNETES Architecture





Master, Node



주요 구성 요소

Master :

마스터 서버에는 다양한 제어 모듈이 위치 함.

- **API Server** : 모든 명령은 here를 통함(중요한 설계철학)

운영자 및 내부 노드(node 내의 kubelet)와 통신, Restful API 형태로 운영자 및 kubelet이 접촉할 수 있게 함

- **Controller Manager** : 다양한 컨트롤러 (복제/배포/상태/ ..)를 관리하고 API Server와 통신하여 작업을 수행

- **Scheduler** : 요청된 컨테이너화된 workload를 어느 노드에 배치시킬지를 결정, Kubelet과 정보교류를 통해 각노드의 상황을 바탕으로 결정함

- **etcd** : 가볍고 빠른 분산형 key-value 저장소 / 설정 및 상태정보를 저장

Node :

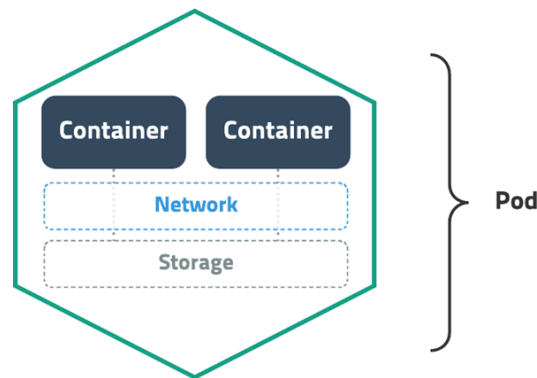
PoD로 불리는 기본 컨테이너(예전이름 minion) 가 운영되는 서버

kubectl

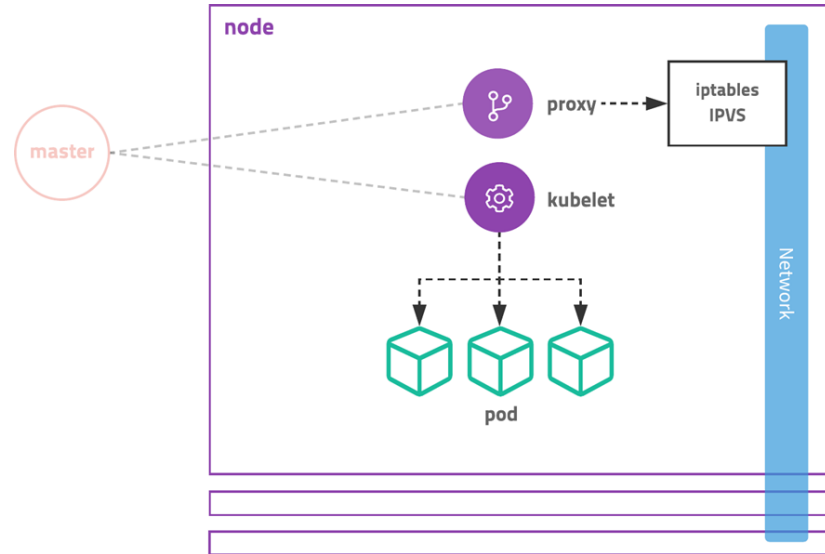
: 명령어 도구, 즉 명령어를 API server의 restful API로 변환

Node > PoD > Container

- ✦ PoD: K8S에서 컨테이너가 동작되는 기본자원을 PoD란 이름으로 정의함
- ✦ PoD 안에 주로 한 개의 컨테이너만 위치하나 직접 종속된 한 두개의 컨테이너를 더 둘 수도 있음
- ✦ PoD에 적정한 스토리지, 네트워킹기능을 할당할 수 있고, 한 PoD 안의 컨테이너들은 이를 공유함



Node 안의 주요 Agent들 : Kubelet, Kube Proxy, Container Engine



kubelet: 마스터의 API Server와 통신하며 Pod를 생성하고 상태를 관리, API Server로 부터 확인한 본 노드의 상태(요구상태)를 실현하기위해 동작함

Proxy: 노드내의 특정 Container와 접촉하고자 할경우 프록시를 통해 정확한 IP 주소등을 확인하여 접속할 수 있게 지원함

Container Engine: 노드내의 가상화기반으로 Docker 뿐만 아니라 현재는 다양한 컨테이너 엔진이 지원됨