Docker & k8s : 컨테이너 기술에 대한 설명과 등장 배경

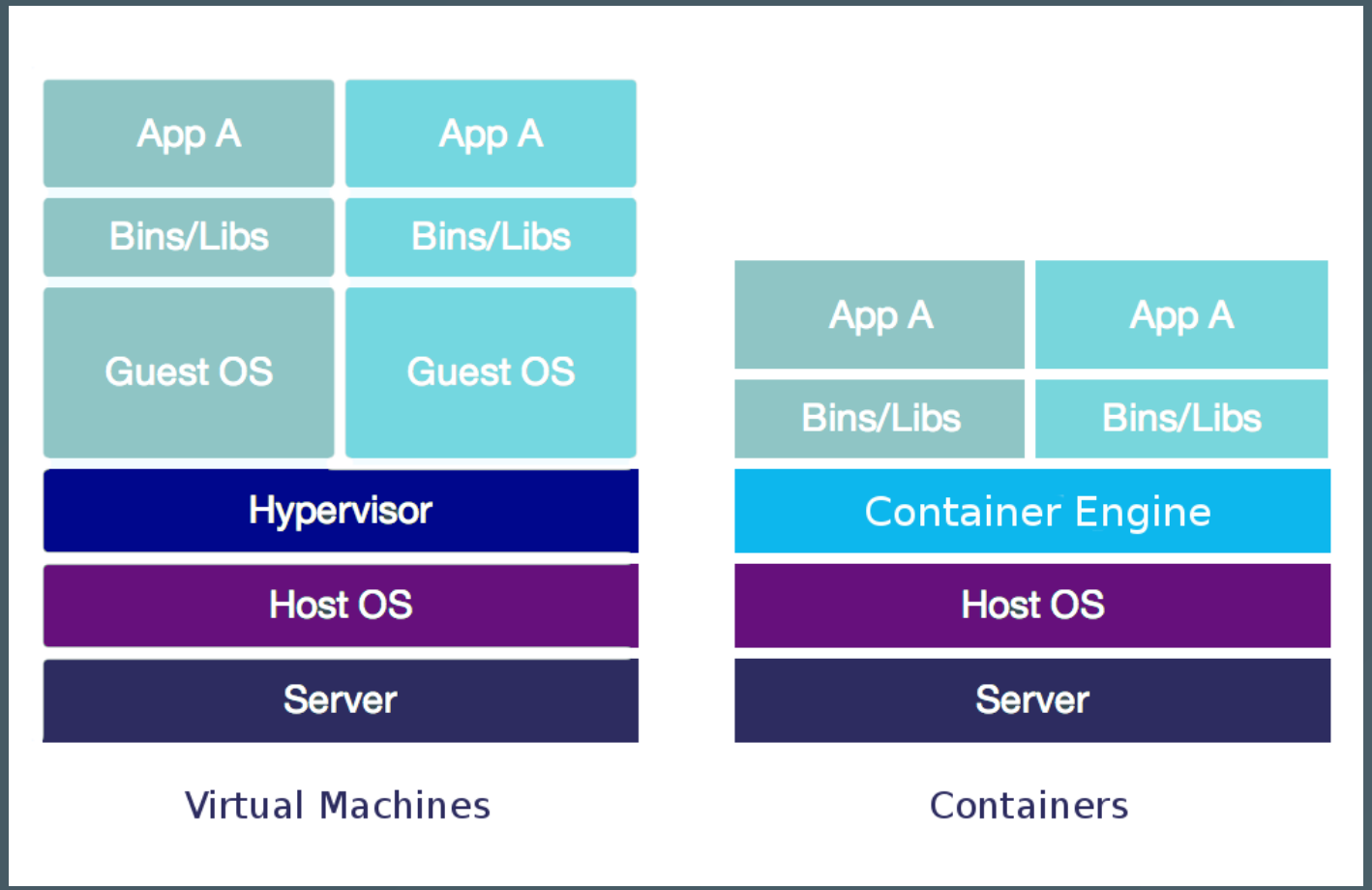
ICT기획부 선임 임형우

자신의 앱을 배포하려는 개발자 L씨...

1. 미묘하게 다른 개발 환경으로 인한 버그 발생 - 'it works on my machine'
2. 다른 OS에서 테스트 - 서로 다른 의존성 관리 방법 및 패키지 종류
3. while (**install - 확인 - 확인 - 수락 - ...**)
4. 장비 값 부담

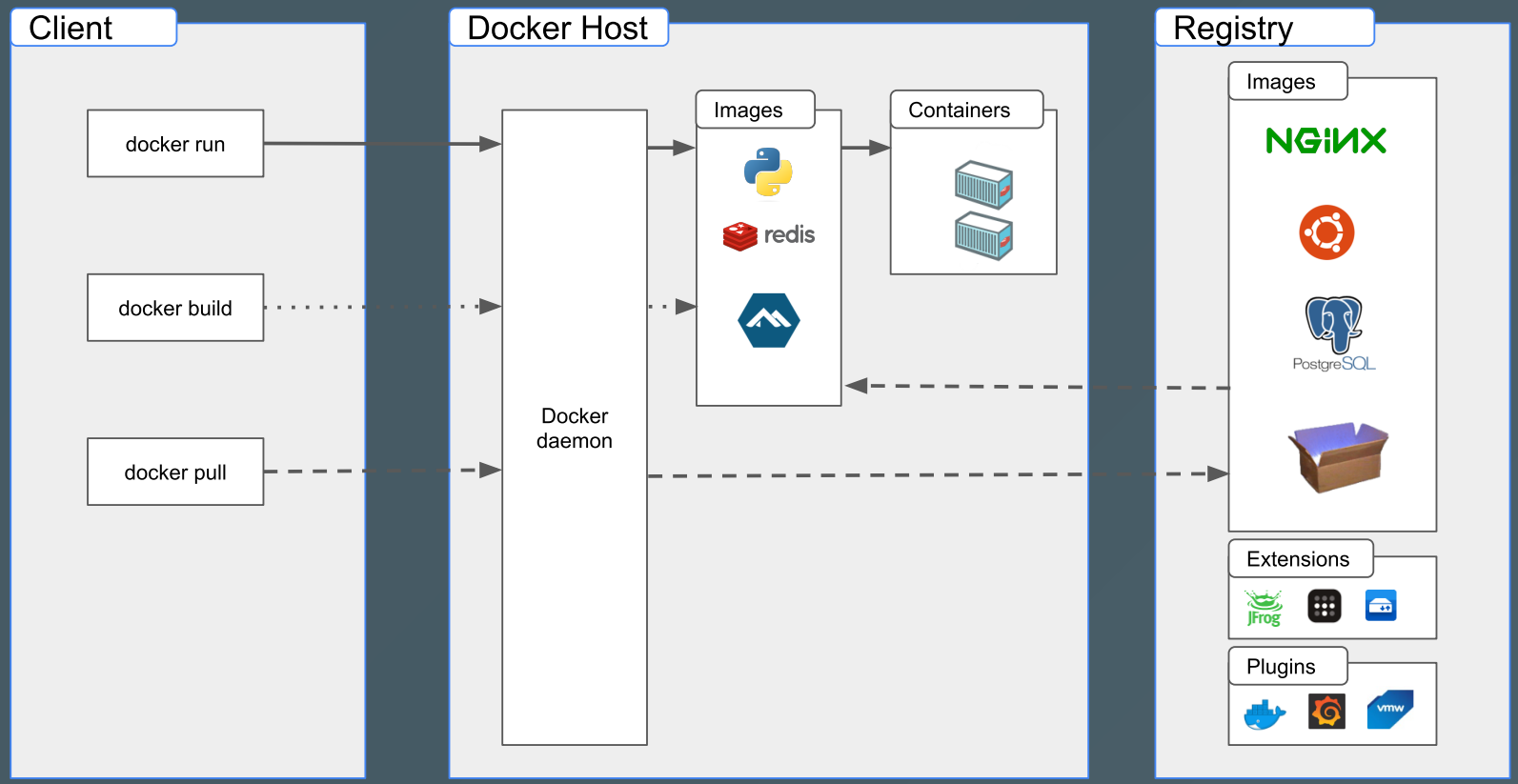
위의 문제를 어떻게 해결하면 좋을까요?

컨테이너


App A
App A
Bins/Libs
Bins/Libs
Guest OS
Guest OS
Hypervisor
Host OS
Server
Virtual Machines
App A
App A
Bins/Libs
Bins/Libs
Container Engine
Host OS
Server
Containers

docker®

도커 아키텍처


Client
docker run
docker build
docker pull
Docker Host
Docker daemon
Images
redis
Containers
Registry
Images
NGINX
PostgreSQL
Extensions
JFrog
Plugins
vmw

Docker Objects

- 이미지
- 컨테이너
- 볼륨
- 네트워크

이미지

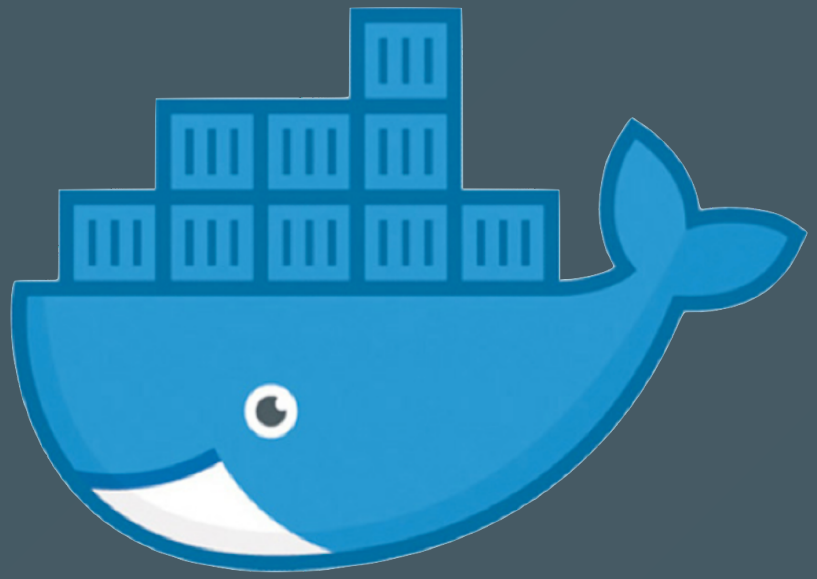
```dockerfile
# syntax=docker/dockerfile:1

FROM node:18-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
EXPOSE 3000
```

- **read-only**, 레이어 구조, 쉬운 공유, 빌드에 사용됨, 스냅 샷

컨테이너

Docker Image

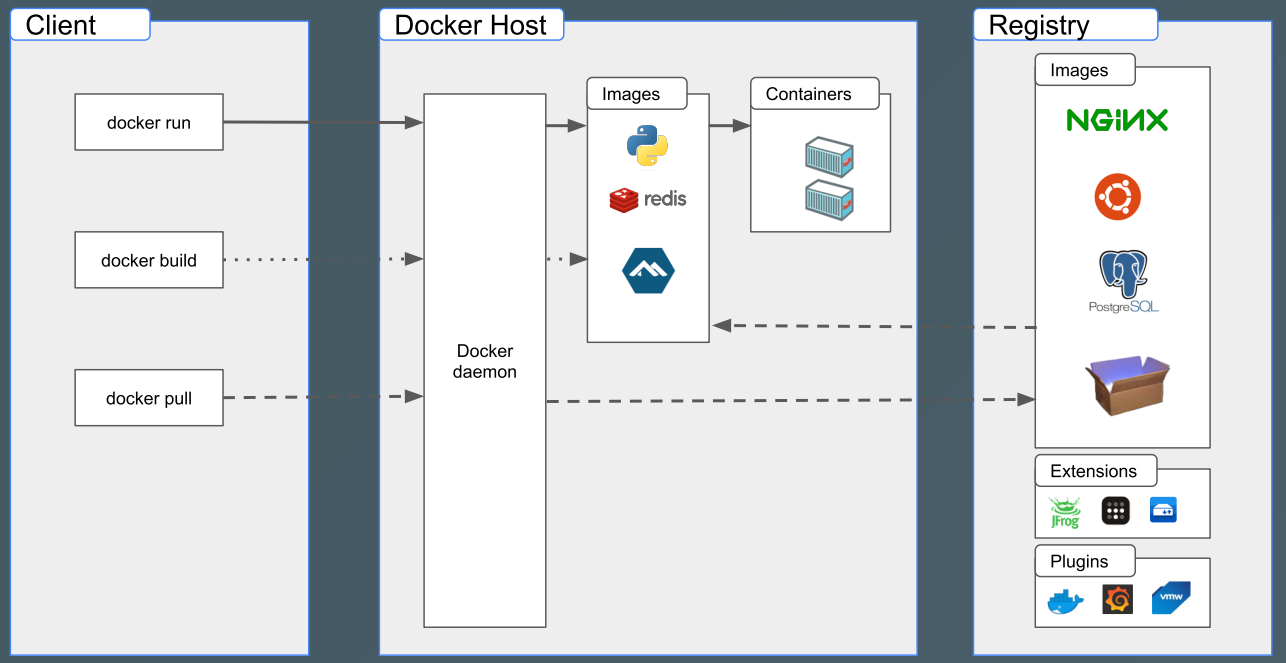


Docker Container

- 이미지의 인스턴스, 일관성, namespaces, cgroup

Recap : 도커 아키텍처

```
$ docker run -i -t ubuntu /bin/bash
```



또 다른 문제에 봉착한 개발자 L씨...

- 로드 밸런싱
- 컨테이너 관리 자동화
- Scale Out의 어려움
- ㅠㅠ

컨테이너 오케스트레이션

컨테이너화 된 애플리케이션에 대한 자동화된 설정, 관리 및 제어 체계

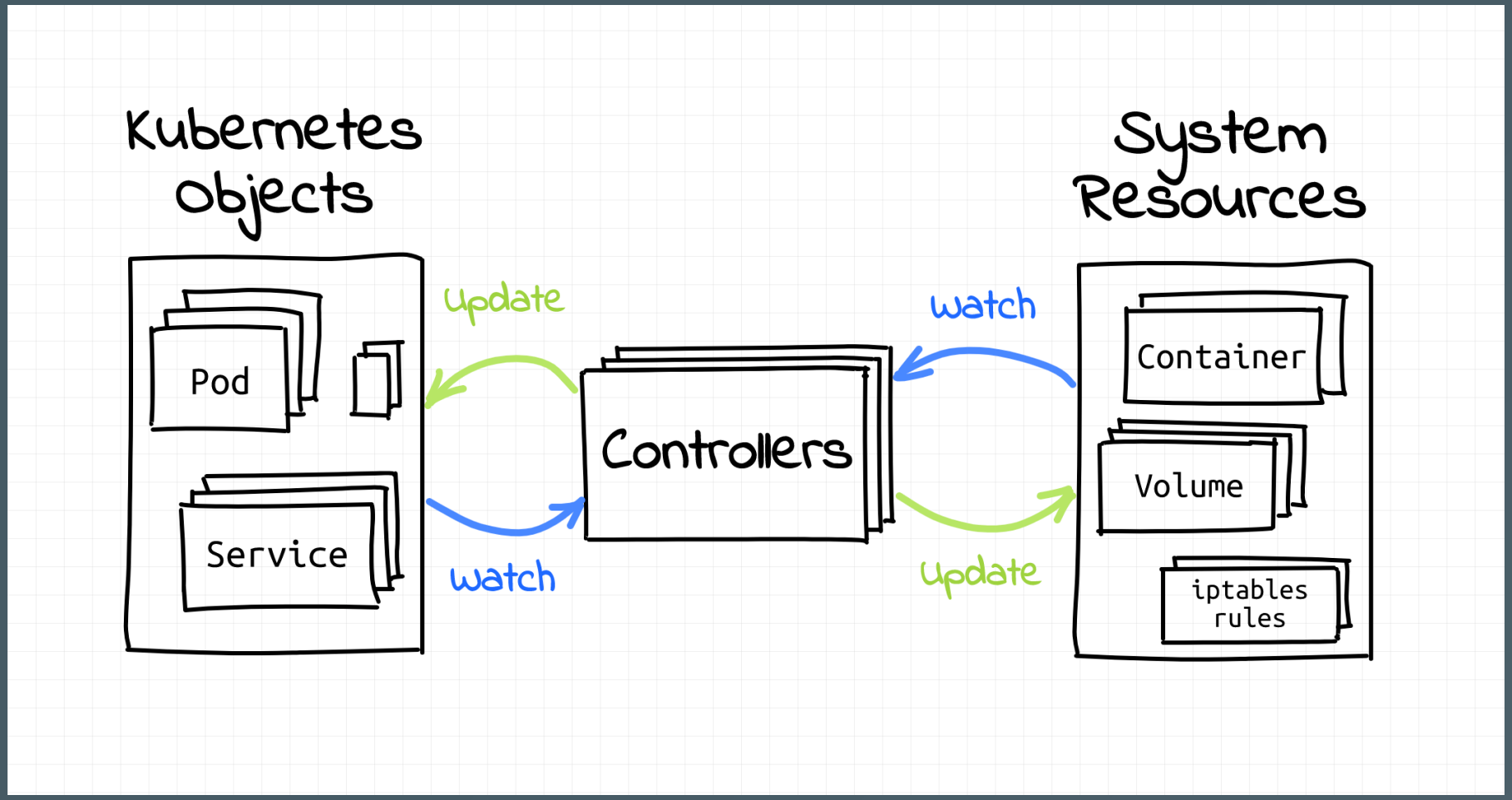
1. 배포 관리
2. 제어 및 모니터링
3. 스케일링
4. 네트워킹

쿠버네티스 핵심 설계 사상

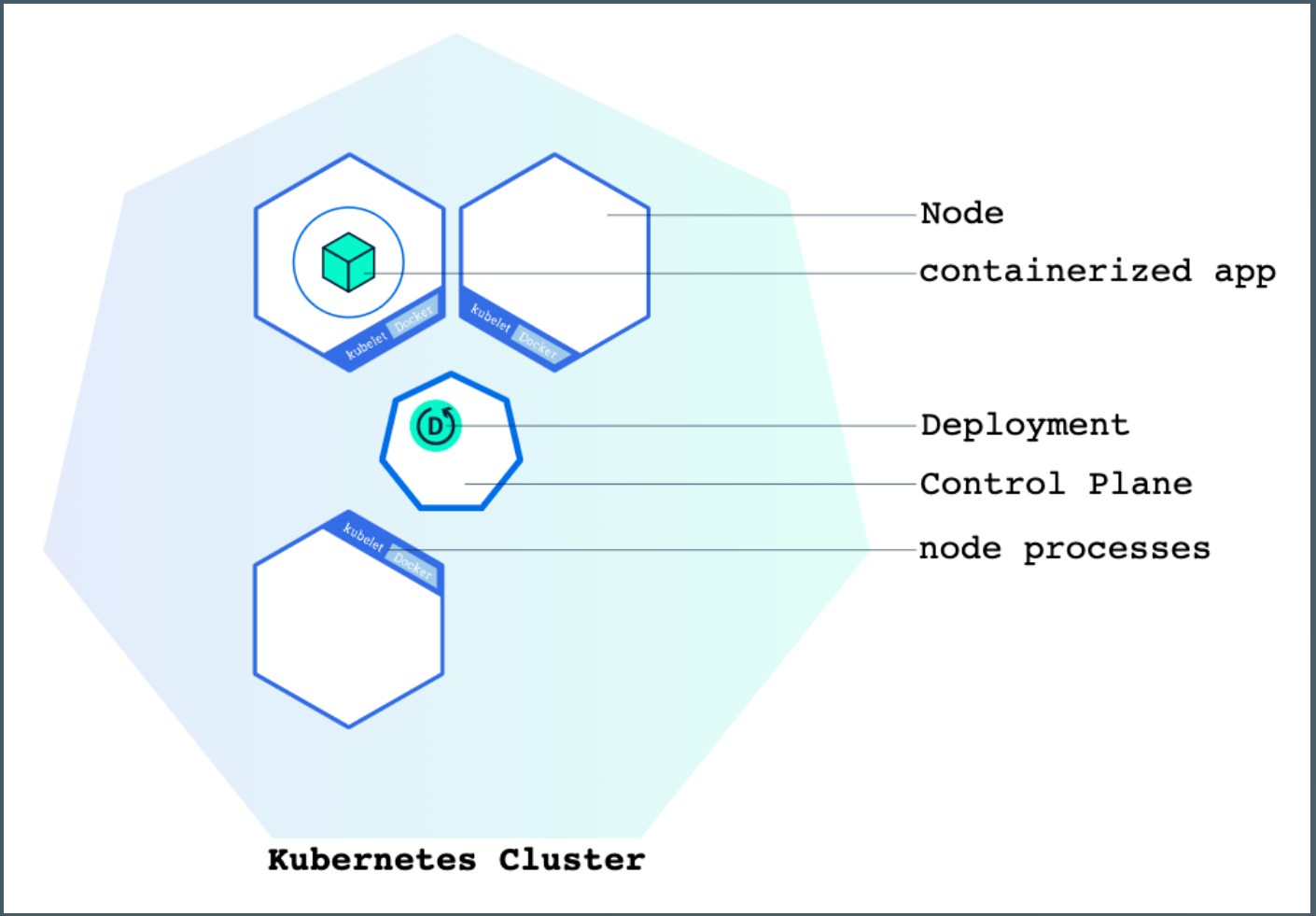
1. 선언적 구성 기반의 배포 환경
2. 기능 단위의 분산
3. 클러스터 단위 중앙 제어
4. 동적 그룹화
5. API 기반 상호작용

선언적 구성 기반의 배포 환경

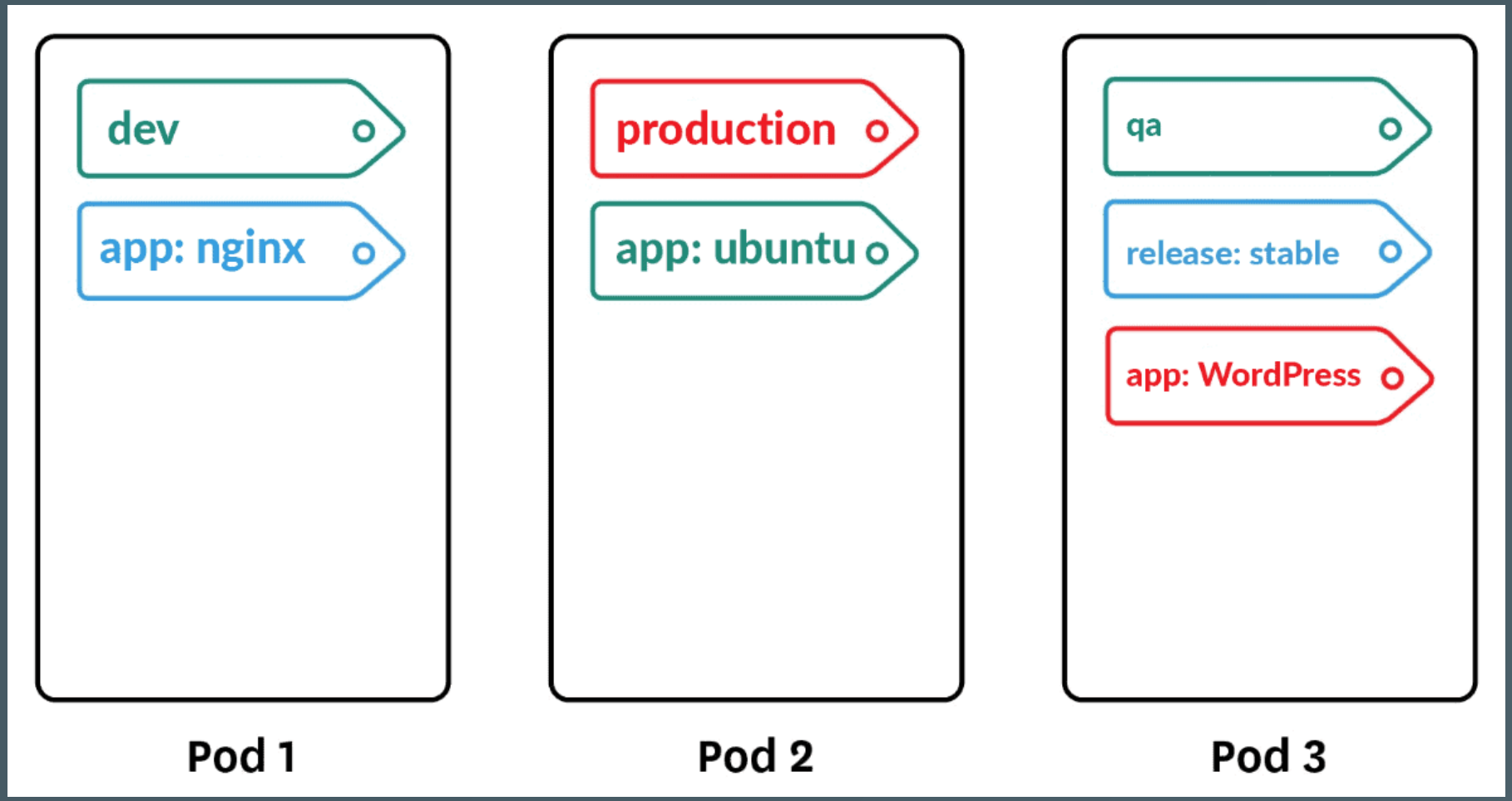
기능 단위의 분산


Kubernetes objects
Pod
Service
Update
watch
Controllers
watch
Update
System Resources
Container
Volume
iptables rules

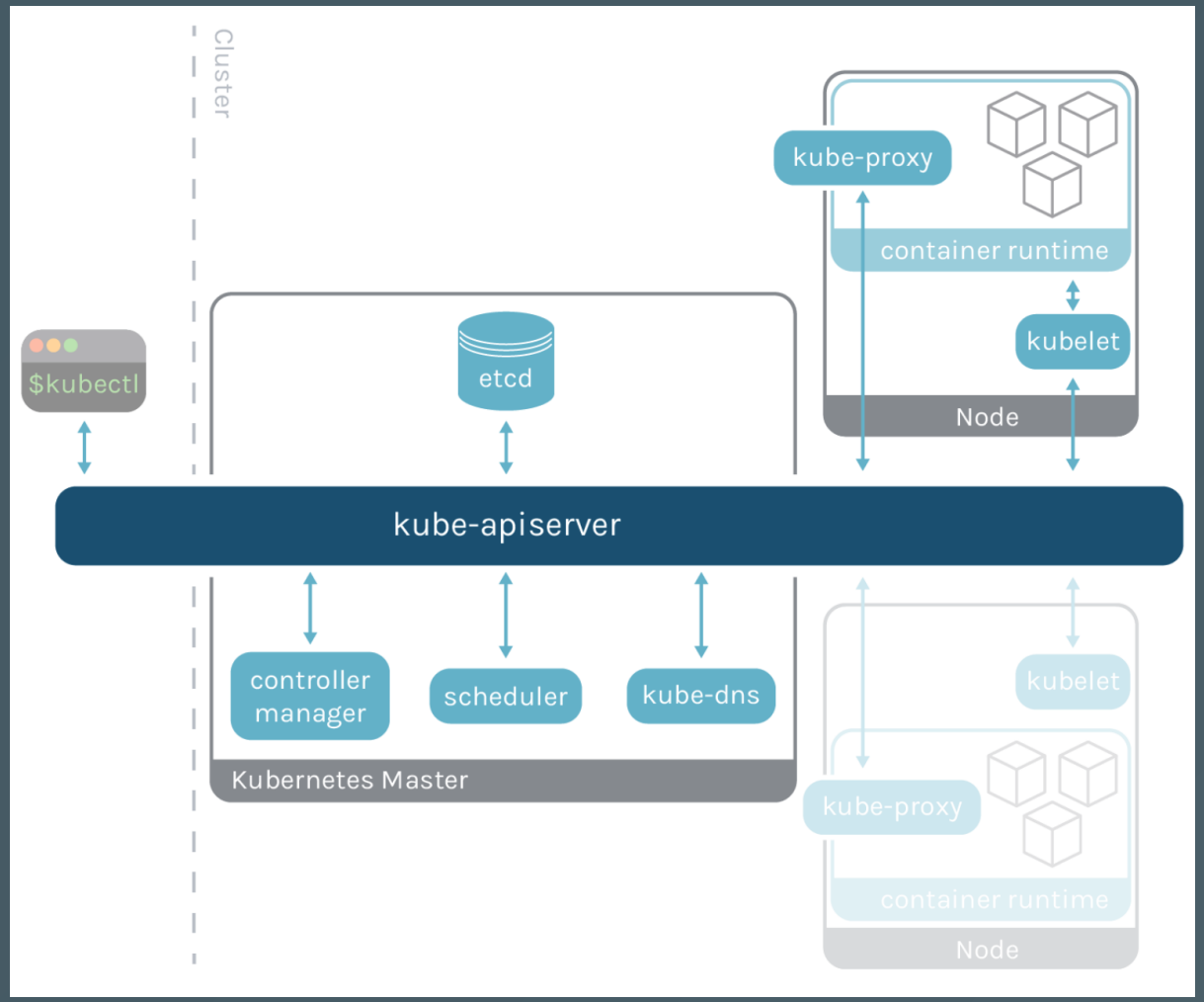
클러스터 단위 중앙 제어


kubelet
Docker
kubelet
Docker
kubelet
Docker
Node
containerized app
Deployment
Control Plane
node processes
Kubernetes Cluster

동적 그룹화


dev
app: nginx
production
app: ubuntu
qa
release: stable
app: WordPress
Pod 1
Pod 2
Pod 3

API 기반 상호작용


Cluster
kube-proxy
container runtime
kubelet
Node
$kubectl
etcd
kube-apiserver
controller manager
scheduler
kube-dns
Kubernetes Master
kubelet
kube-proxy
container runtime
Node

EOF

- https://docs.docker.com/
- https://kubernetes.io/ko/docs/concepts/
- https://seongjin.me/kubernetes-core-concepts/
- https://m.blog.naver.com/shakey7/221600166205