Journey to achieving Cloud Native at Kakao

정기훈(lev.jung@kakaocorp.com)
클라우드네이티브파트, 카카오

발표자 소개

정기훈

#SoftwareEngineer#Cloud#LargeScale#Networking#Automation#CICD#SRE

현) 클라우드네이티브파트 파트장, 카카오
전) SW R&D Center, 삼성전자

01 What is Cloud Native?

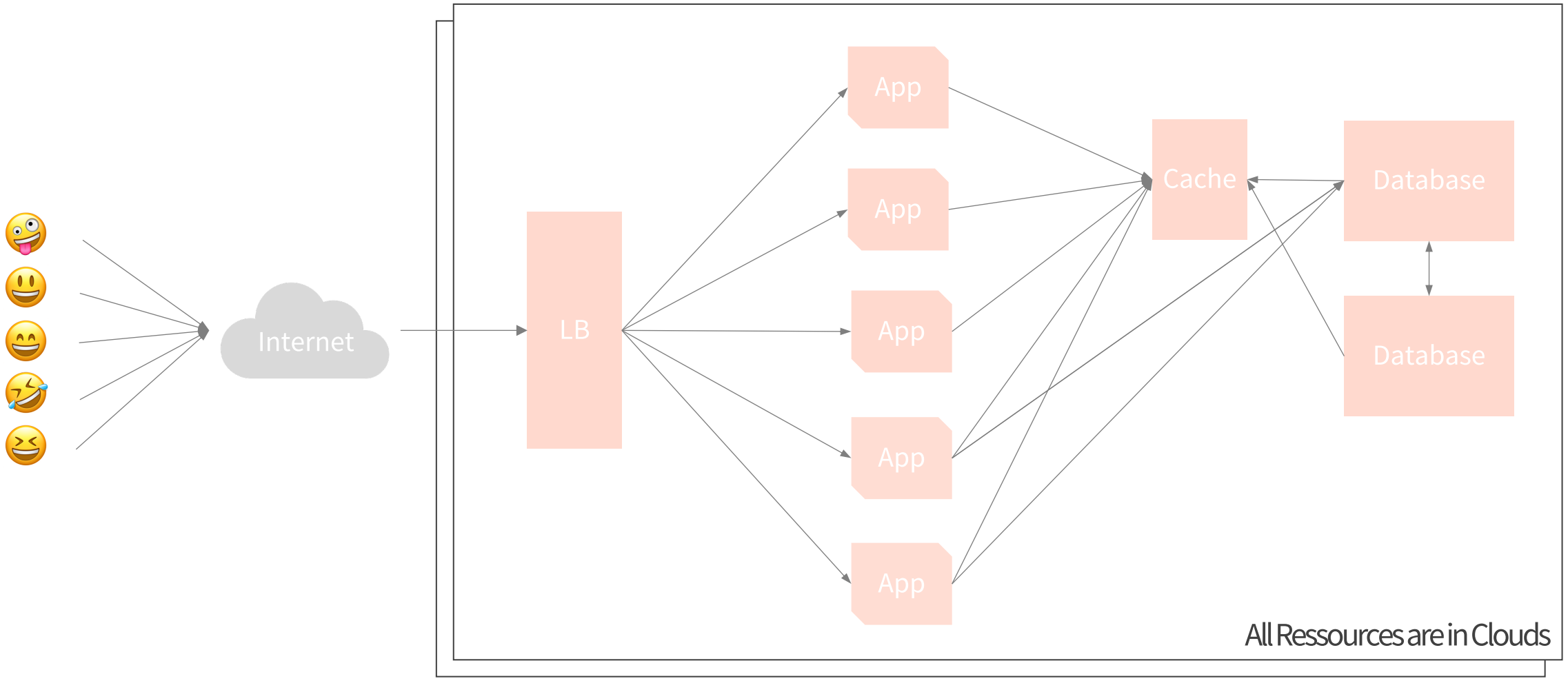
Cloud Native Definition from CNCF

Empower organizations to build and run scalable applications in clouds.

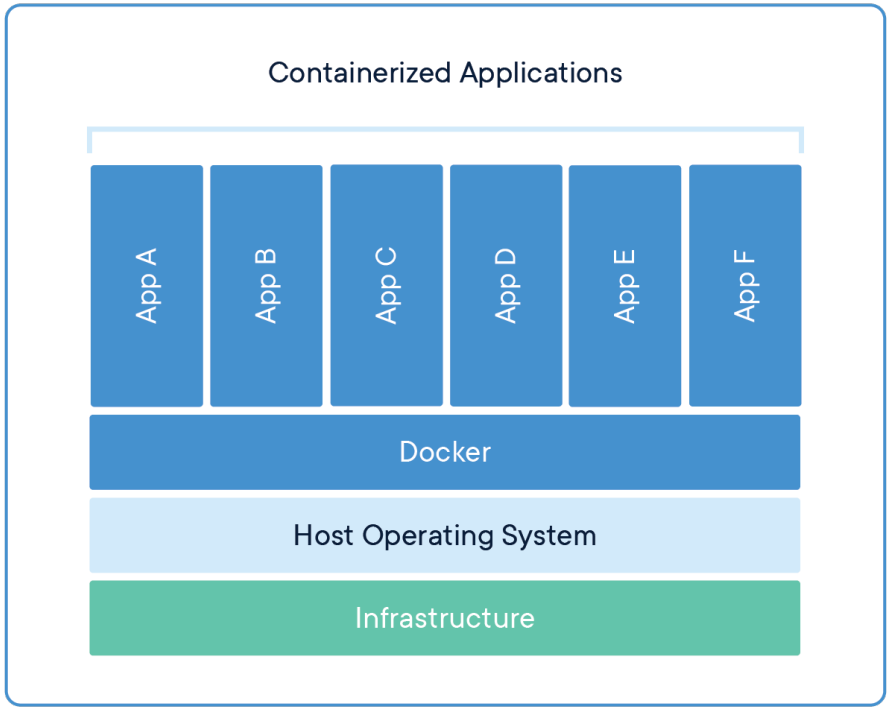
Exemplify approach:
- containers
- microservices
- service meshes
- immutable infrastructure
- declarative APIs

https://github.com/cncf/toc/blob/master/DEFINITION.md

Scalable Application in clouds

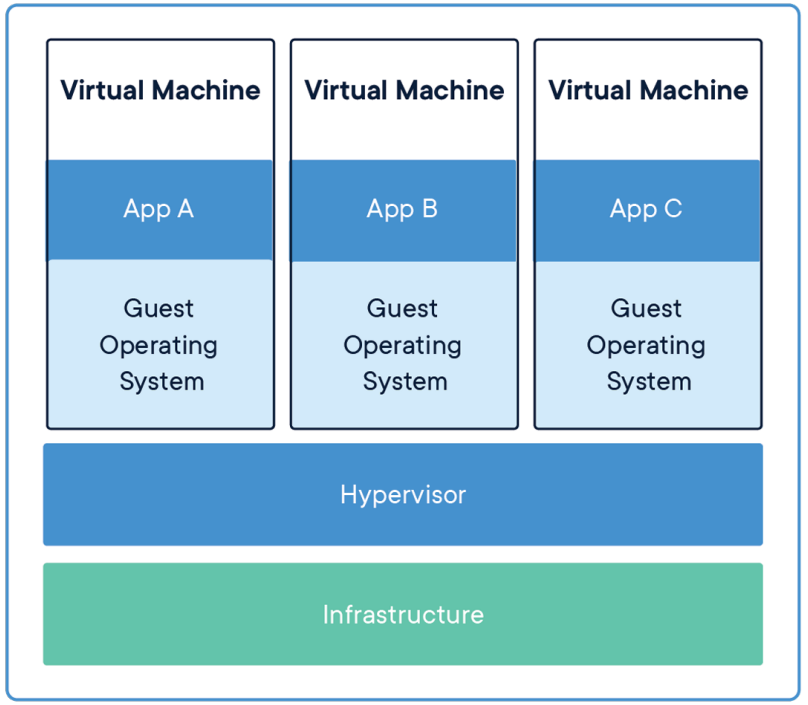


Container



Container

Host Kernel을 공유
User Space 격리 제공
레이어 기반의 이미지 구성

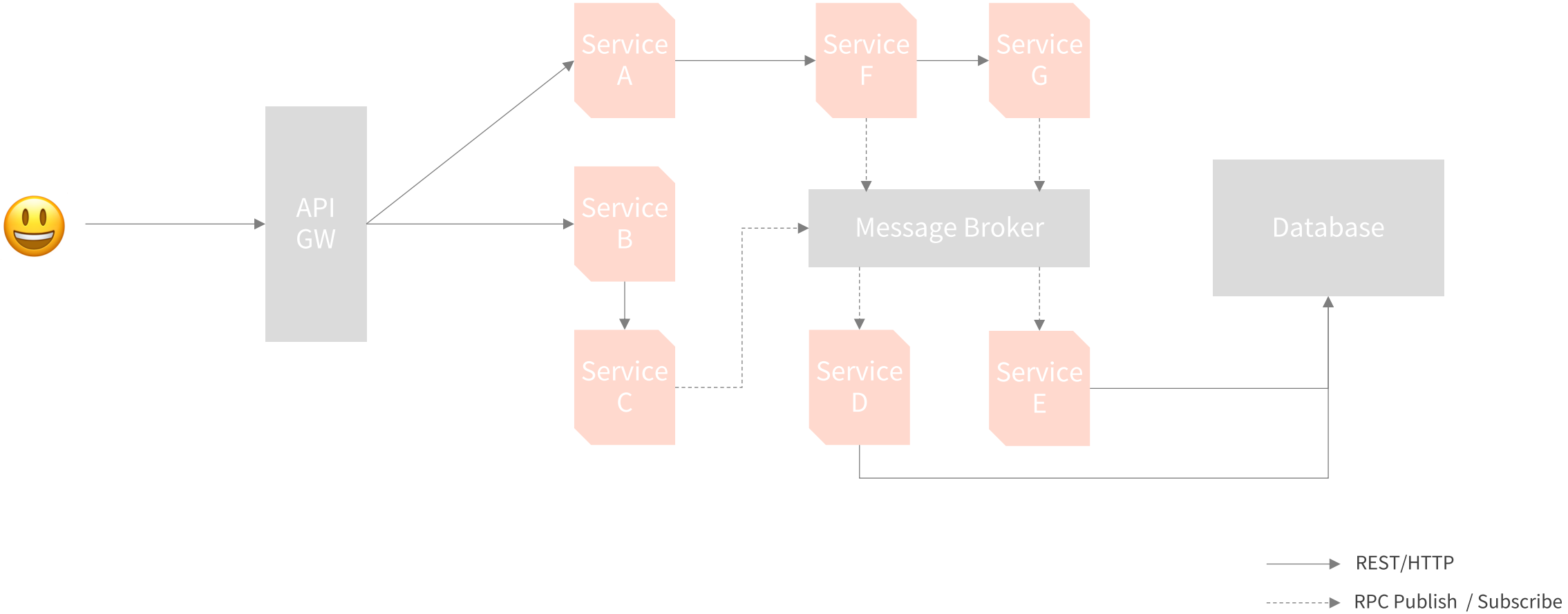


Virtual Machine

Host Kernel과 완전 격리 가능
Guest OS의 선택이 가능
디스크 기반의 이미지 구성

https://www.docker.com/resources/what-container

Microservice Architecture



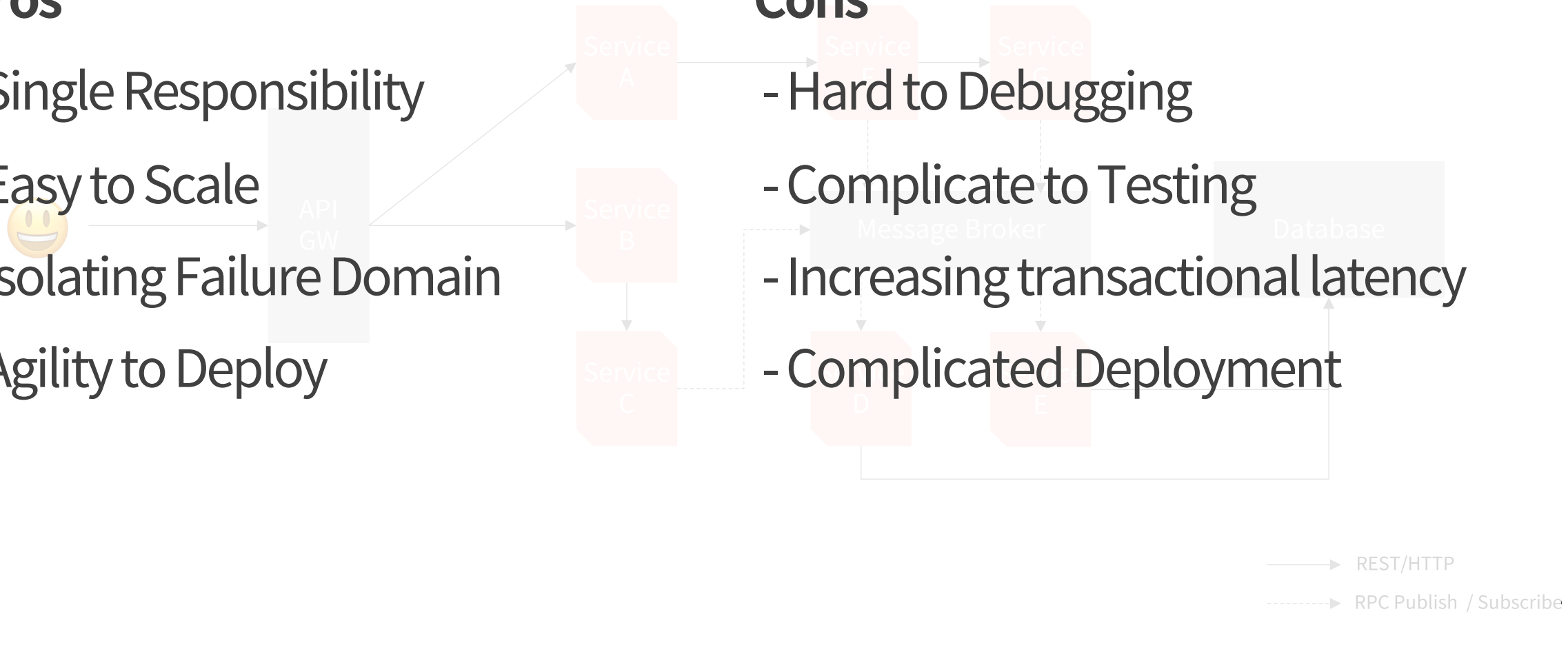
Microservice Architecture

Pros

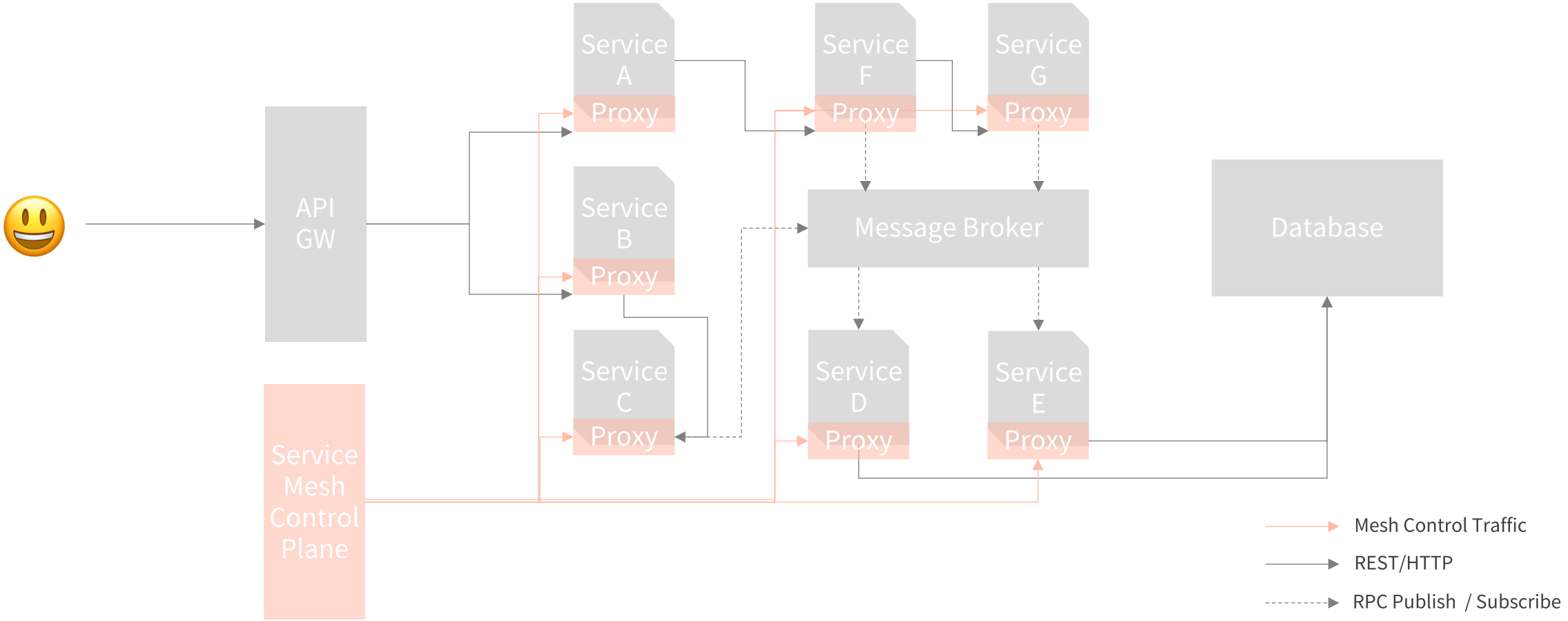
- Single Responsibility
- Easy to Scale
- Isolating Failure Domain
- Agility to Deploy

Cons

- Hard to Debugging
- Complicate to Testing
- Increasing transactional latency
- Complicated Deployment



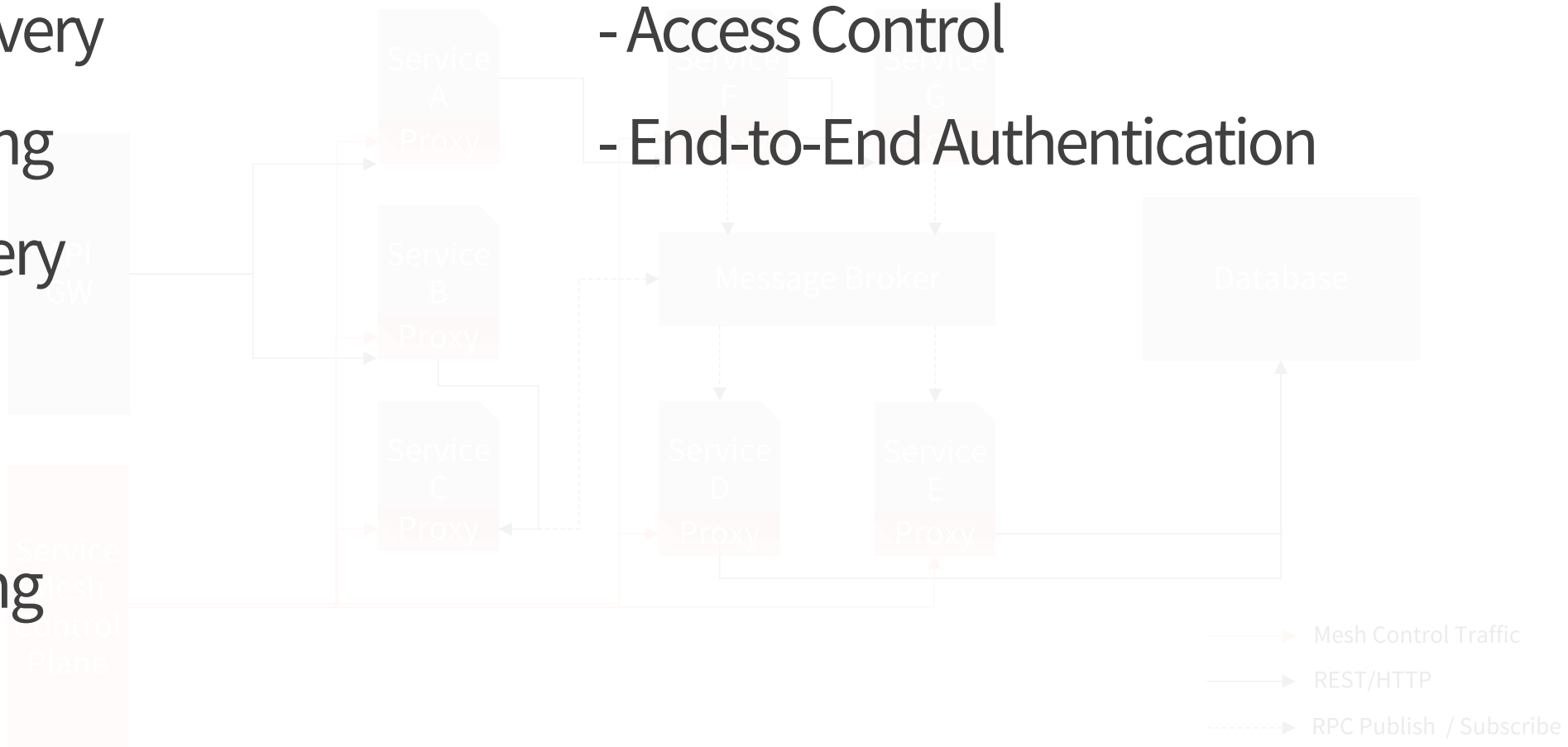
Service Mesh



Service Mesh

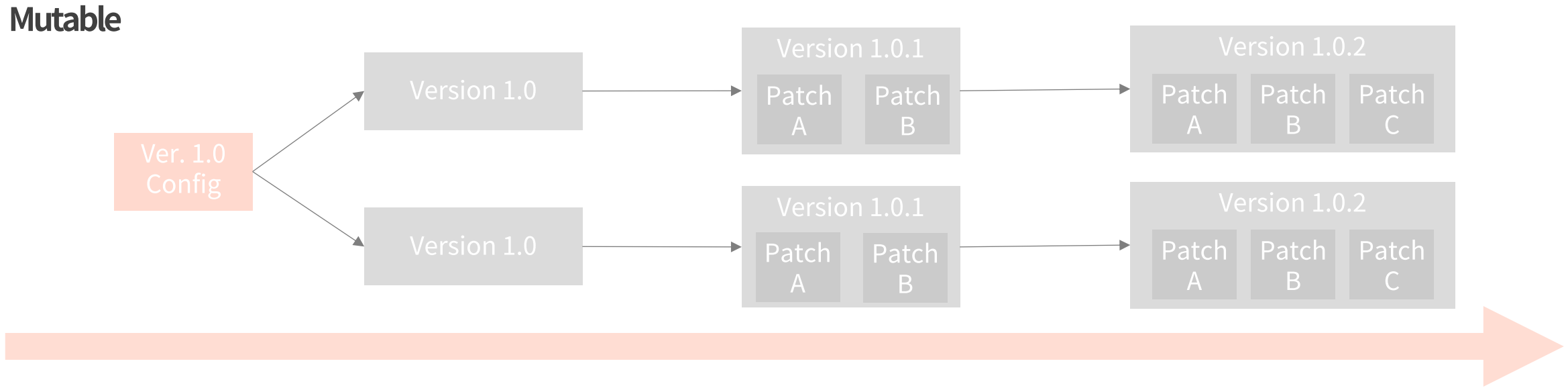
Features

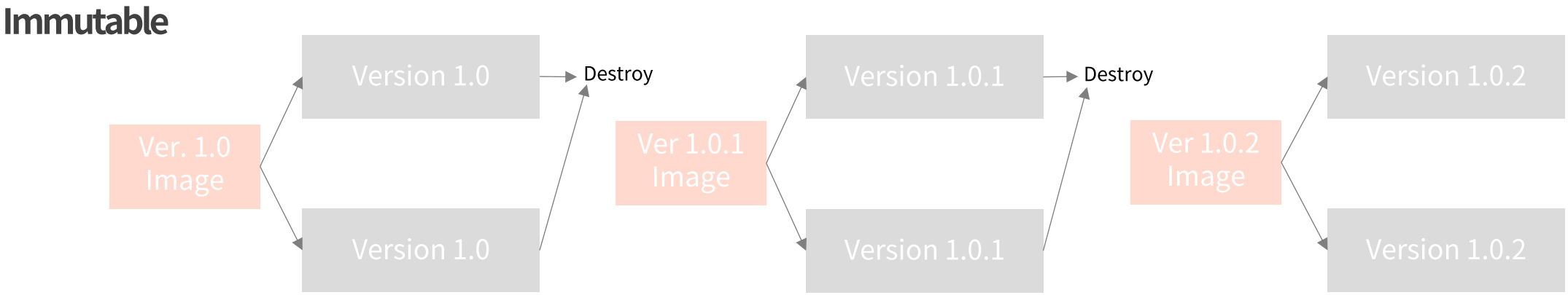
- Service Discovery
- Load Balancing
- Failure recovery
- Metric
- Monitoring
- Traffic Steering
- Access Control
- End-to-End Authentication



https://istio.io/latest/docs/concepts/what-is-istio/#what-is-a-service-mesh

Immutable Infrastructure





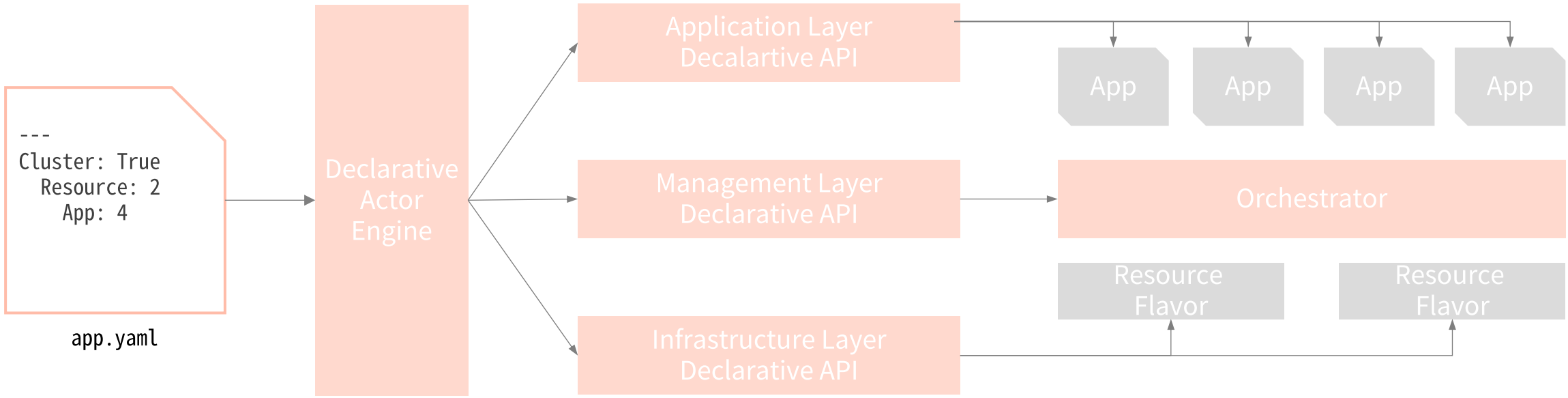
Immutable Infrastructure

Advantage

- No Configuration Drift or Snowflake servers
- Known-good server state and fewer deployment failure
- Consistent staging environment and easy horizon scaling
- Simple rollback and recovery Process

https://www.digitalocean.com/community/tutorials/what-is-immutable-infrastructure#advantages-of-immutable-infrastructure

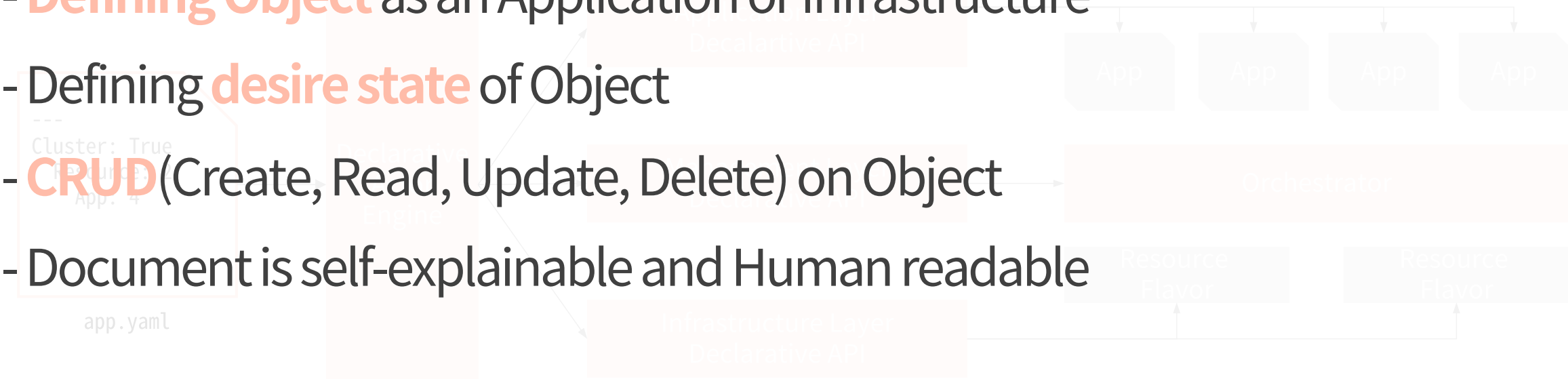
Declarative API



Declarative API

Characteristic

- **Defining Object** as an Application or Infrastructure
- Defining **desire state** of Object
- **CRUD**(Create, Read, Update, Delete) on Object
- Document is self-explainable and Human readable

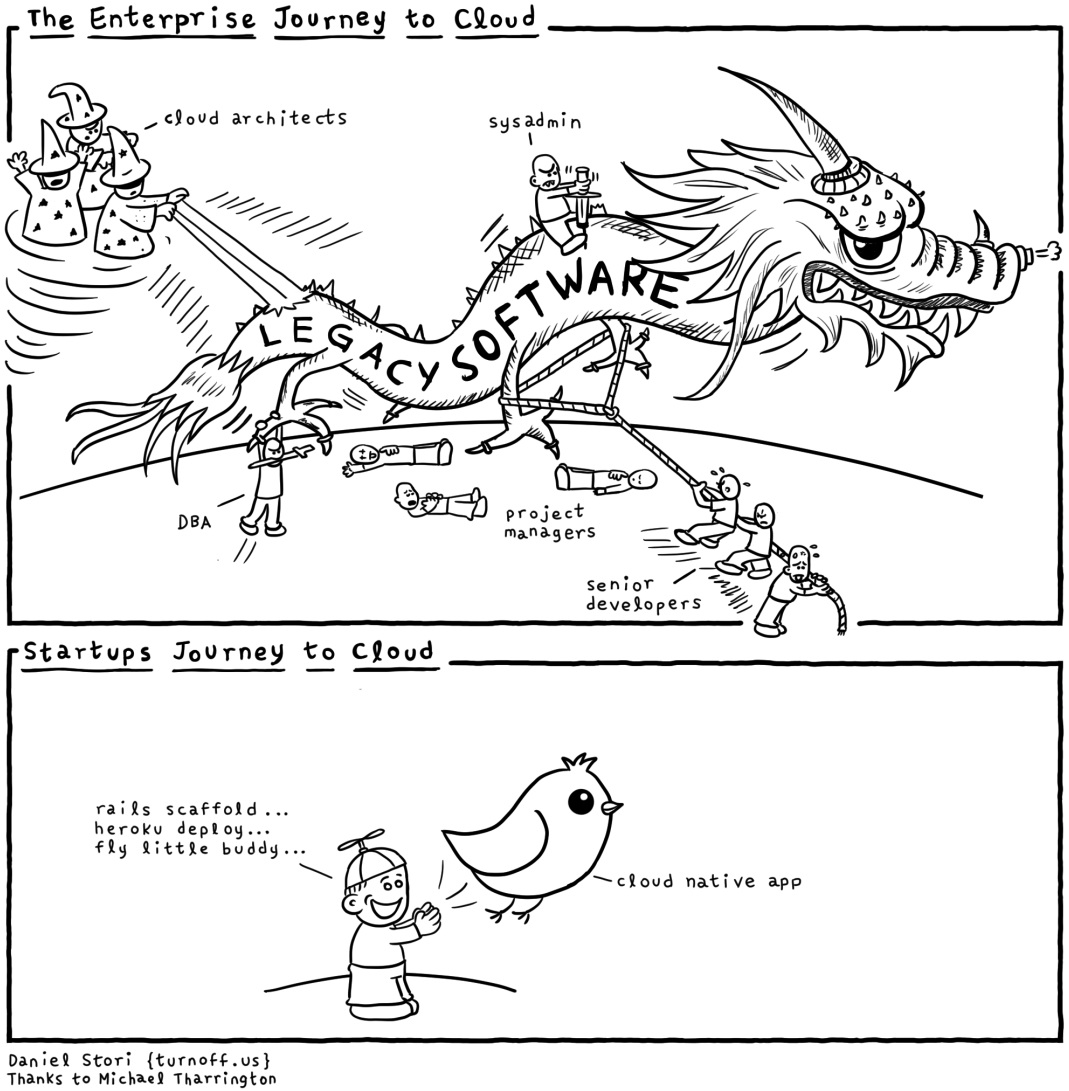


Additionally, Twelve Factors

1. **Codebase**
   One codebase tracked in revision control
2. **Dependencies**
   Explicitly declare and isolate dependencies
3. **Config**
   Store config in the environment
4. **Backing Service**
   Treat backing services as attached resources
5. **Build, Release, Run**
   Strictly separate build and run stages
6. **Process**
   Execute the app as one or more stateless process
7. **Port Biding**
   Export services via port binding
8. **Concurrency**
   Scale out via the process model
9. **Disposability**
   Maximize robustness with fast startup and graceful shutdown
10. **Dev/prod parity**
    Keep development, staging, and production as similar as possible
11. **Log**
    Treat logs as event streams
12. **Admin processes**
    Run admin/management task as one-off processes

https://12factor.net/

However,



http://turnoff.us/geek/enterprise-vs-startup-journey-to-cloud/

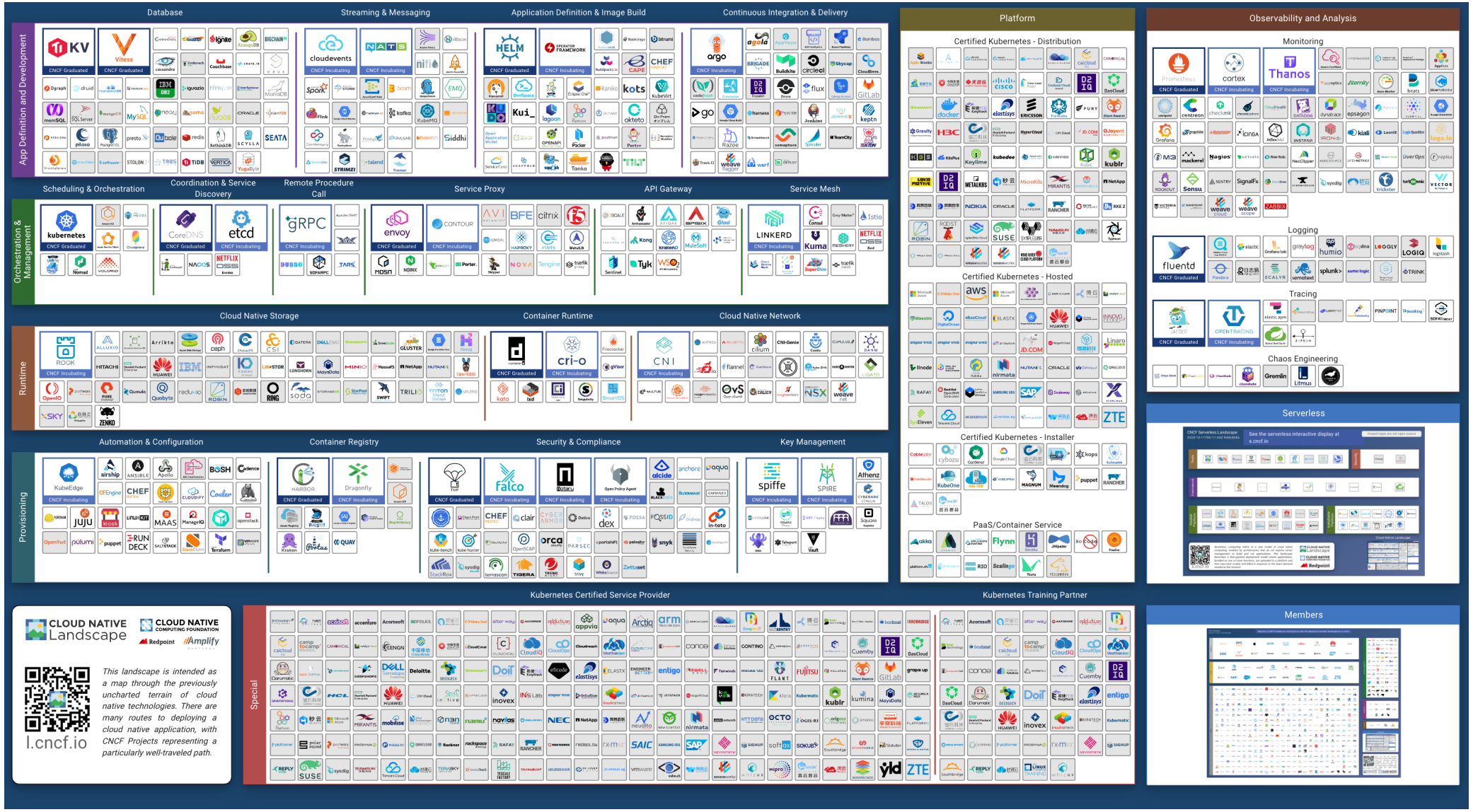
But People say,

“그래서,

이제는 쿠버네티스만 있으면 다 되는 거죠!?”

…

Kubernetes rules them all?



https://landscape.cncf.io

Long Journey

- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing

envoy — CNCF Graduated
CoreDNS — CNCF Graduated
LINKERD — CNCF Incubating

7. DISTRIBUTED DATABASE & STORAGE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding. Rook is a storage orchestrator that integrates a diverse set of storage solutions into Kubernetes. Serving as the "brain" of Kubernetes, etcd provides a reliable way to store data across a cluster of machines. TiKV is a high performant distributed transactional key-value store written in Rust.

Vitess — CNCF Graduated
ROOK — CNCF Incubating
etcd — CNCF Incubating
KV — CNCF Incubating

9. CONTAINER REGISTRY & RUNTIME

Harbor is a registry that stores, signs, and scans content. You can use alternative container runtimes. The most common, both of which are OCI-compliant, are containerd and CRI-O.

containerd — CNCF Graduated
HARBOR — CNCF Incubating
cri-o — CNCF Incubating

JAEGER — CNCF Graduated
OPENTRACING — CNCF Incubating

6. NETWORKING, POLICY, & SECURITY

To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net. Open Policy Agent (OPA) is a general-purpose policy engine with uses ranging from authorization and admission control to data filtering. Falco is an anomaly detection engine for cloud native.

CNI — CNCF Incubating
Open Policy Agent — CNCF Incubating
falco — CNCF Incubating

8. STREAMING & MESSAGING

When you need higher performance than JSON-REST, consider using gRPC or NATS. gRPC is a universal RPC framework. NATS is a multi-modal messaging system that includes request/reply, pub/sub and load balanced queues. CloudEvents is a specification for describing event data in common ways.

gRPC — CNCF Incubating
NATS — CNCF Incubating
cloudevents — CNCF Incubating

10. SOFTWARE DISTRIBUTION

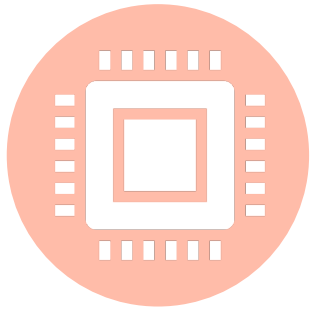
If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.

TUF — CNCF Graduated
notary — CNCF Incubating

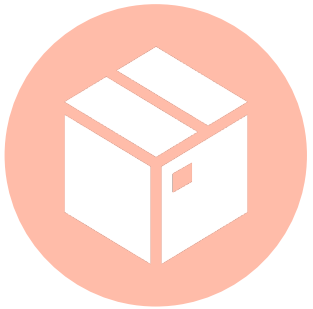
https://github.com/cncf/landscape/blob/master/README.md#trail-map

02 Introducing Kakao Cloud

Some Numbers



PM
0000

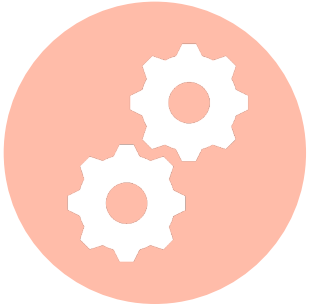


VM
00000

VOLUME
<~ 0000

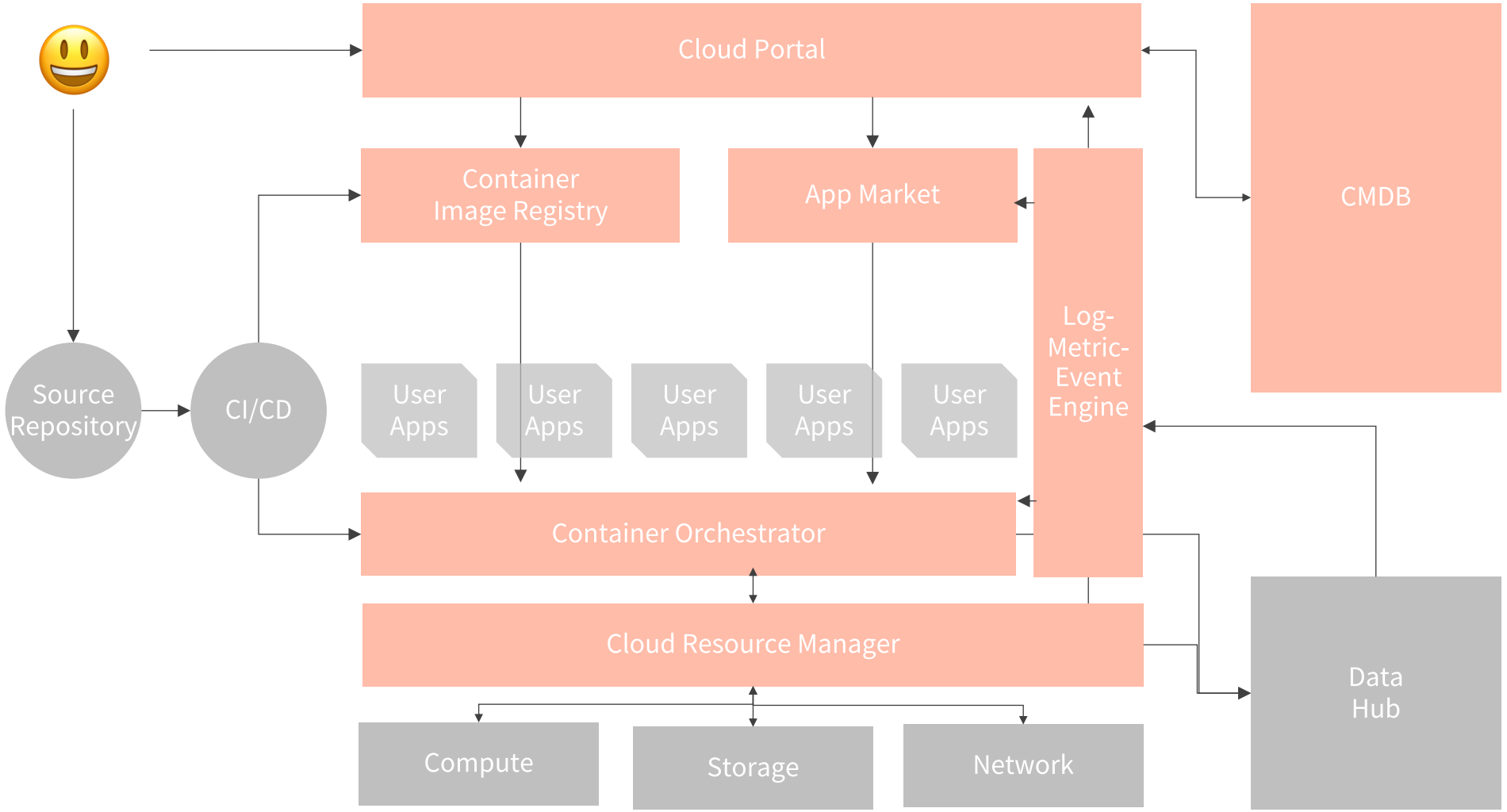
PROJECT
0000

USER
0000

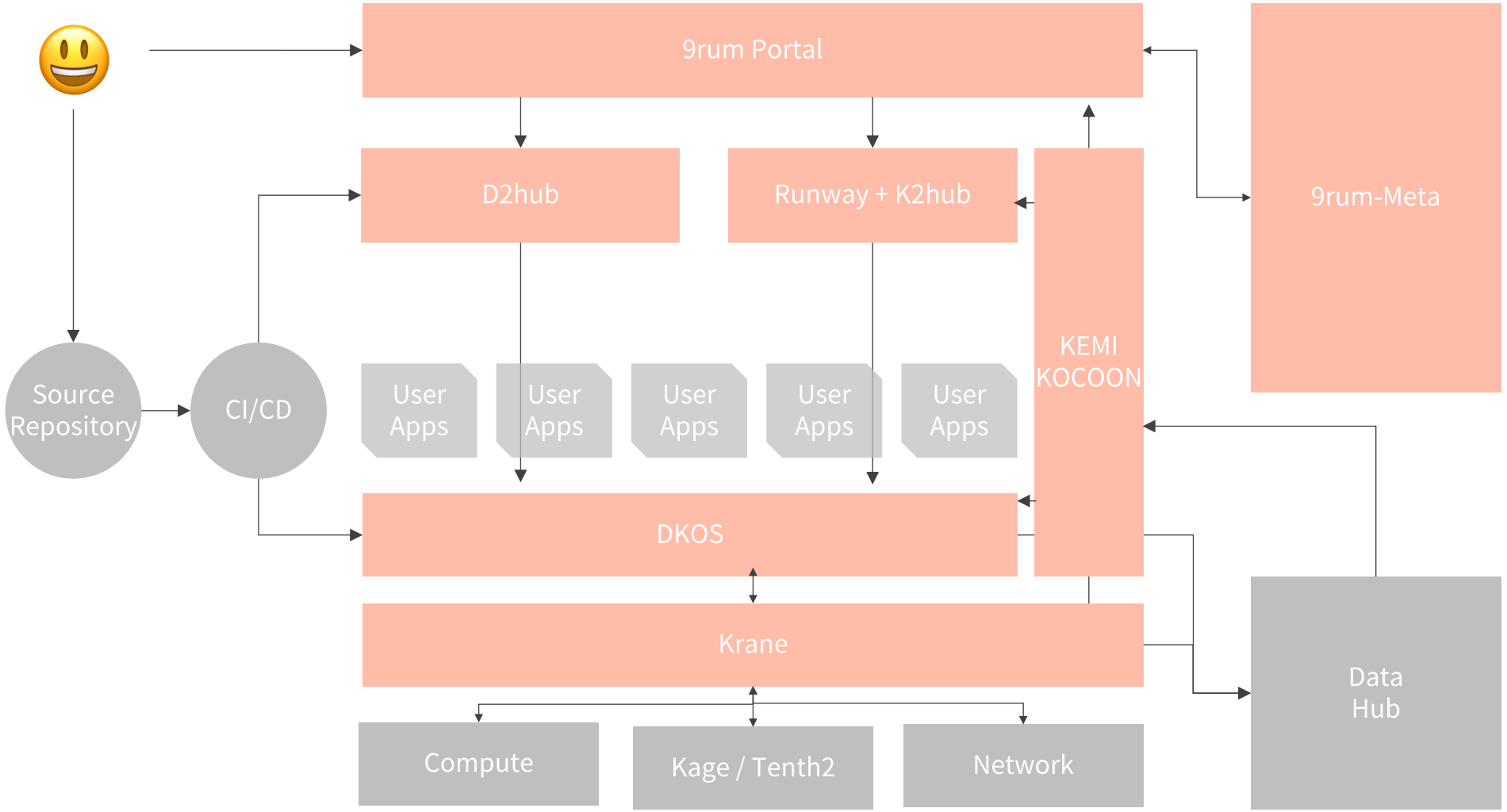


PRODUCTION
0000

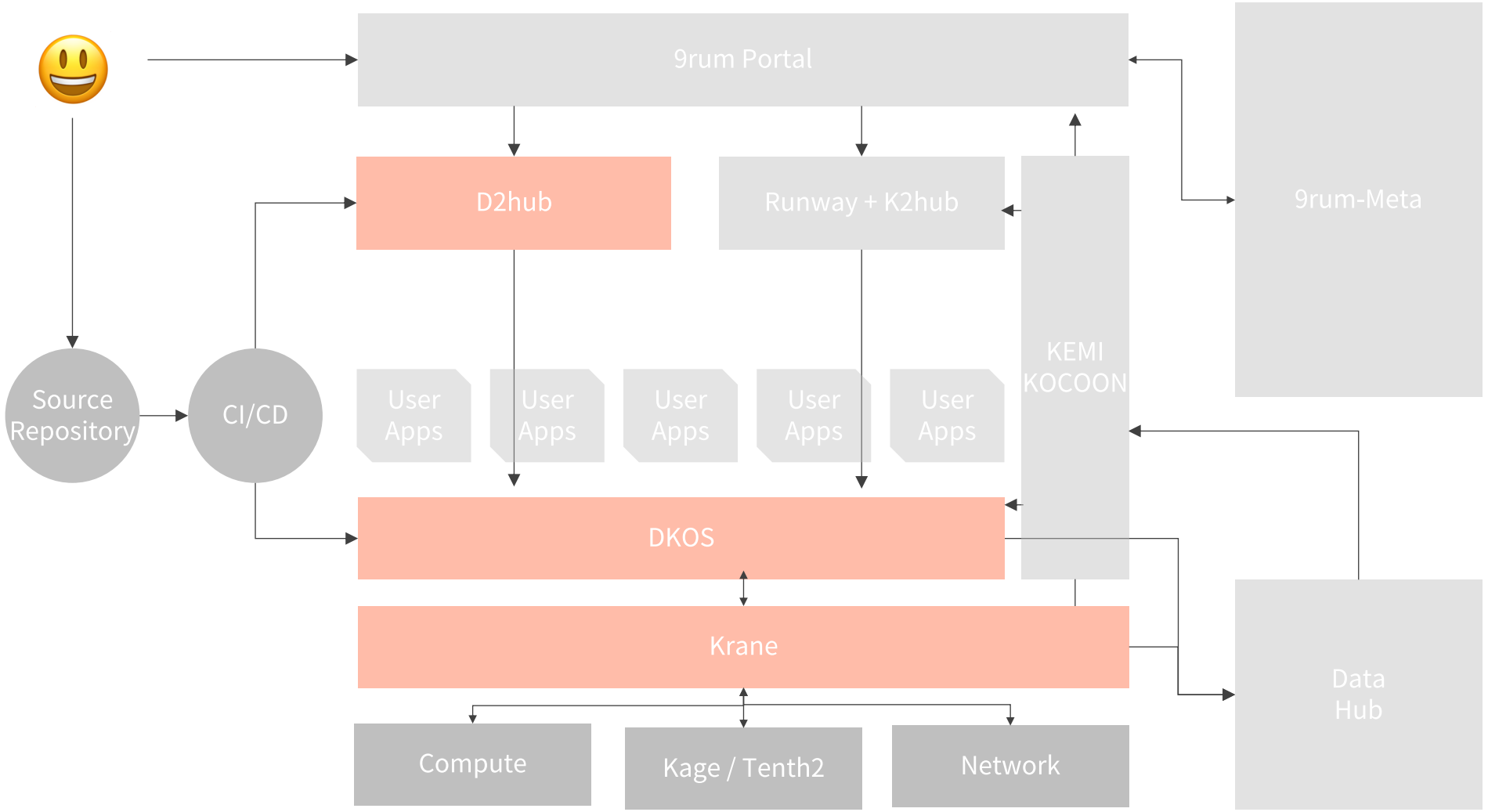
Kakao Cloud Architecture



Kakao Cloud Architecture



Our Product


9rum Portal
D2hub
Runway + K2hub
9rum-Meta
KEMI
KOCOON
Source
Repository
CI/CD
User
Apps
User
Apps
User
Apps
User
Apps
User
Apps
DKOS
Krane
Data
Hub
Compute
Kage / Tenth2
Network

Our Products

Krane

Infrastructure as a Service

OpenStack(stein)

Multiple Regions

L3 Network Model

DKOS

Container Cluster As A Service

Kubernetes

Custom Controller

Kakao Standard Service Platform

D2hub

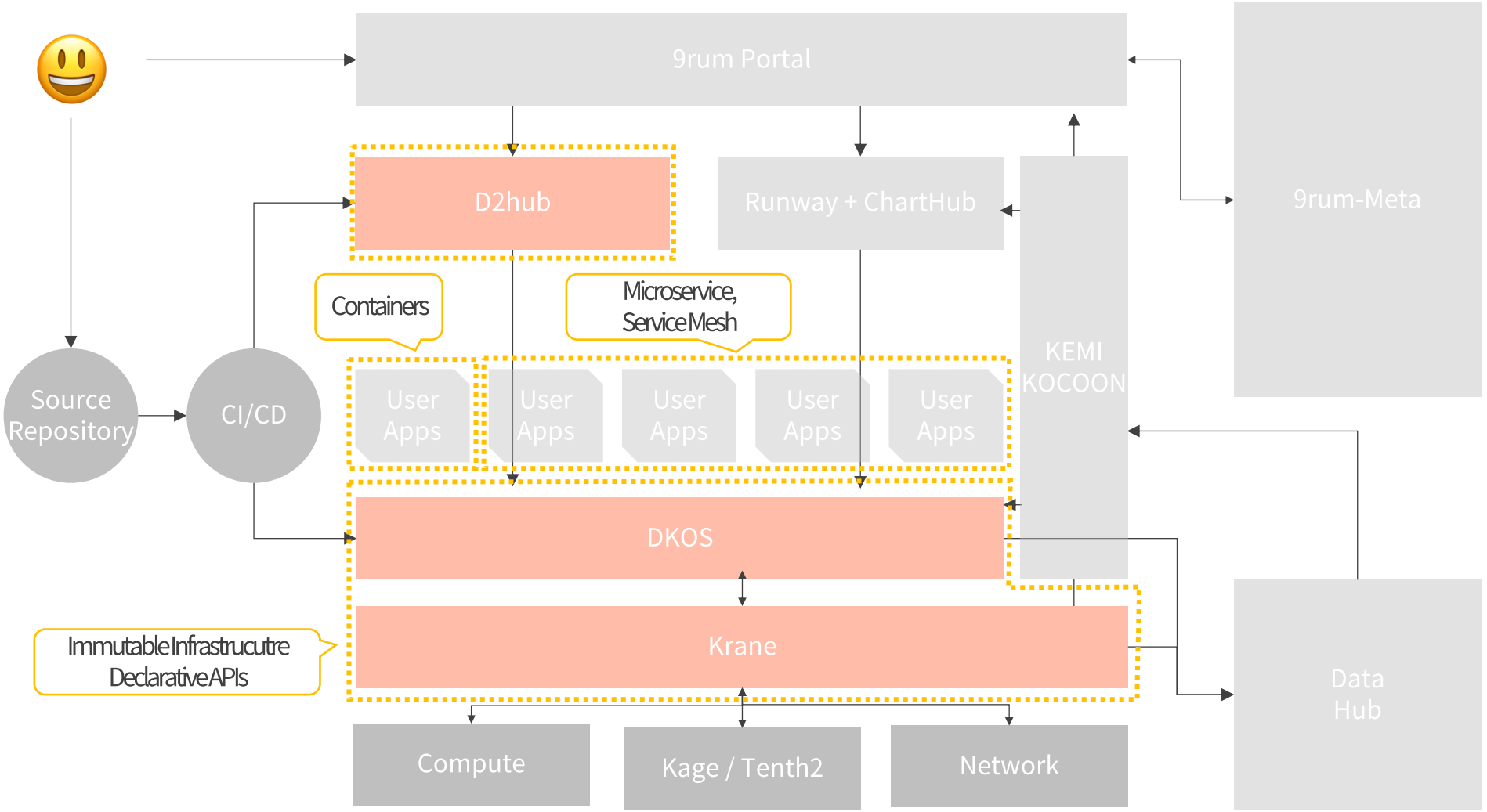
Container Image Registry

Docker Registry

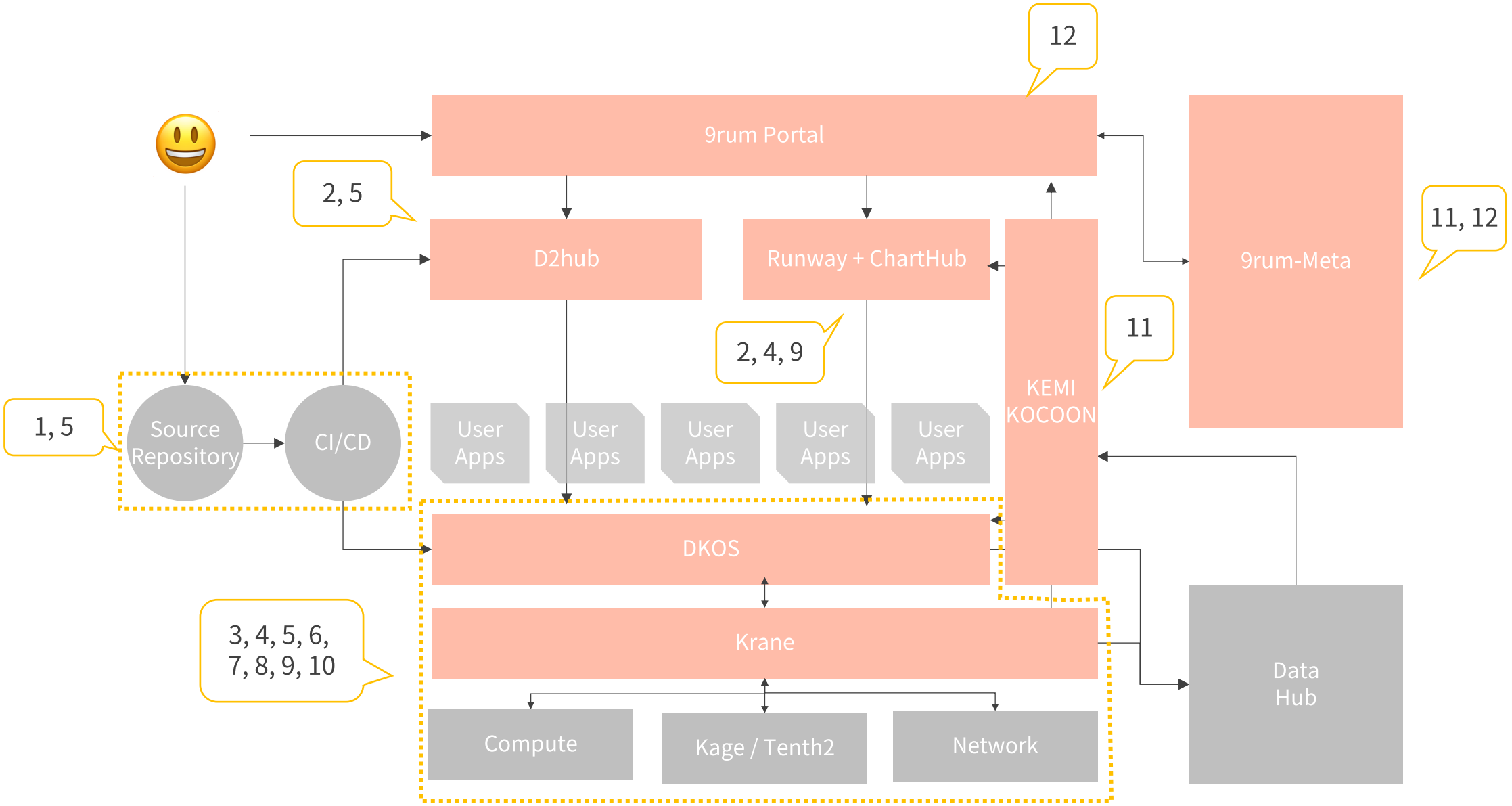
Build and Inspect

Authentication and authorization

Support Cloud Native Technologies on Kakao Cloud



Support 12-factors on Kakao Cloud



03 How Kakao achieved to Cloud Native

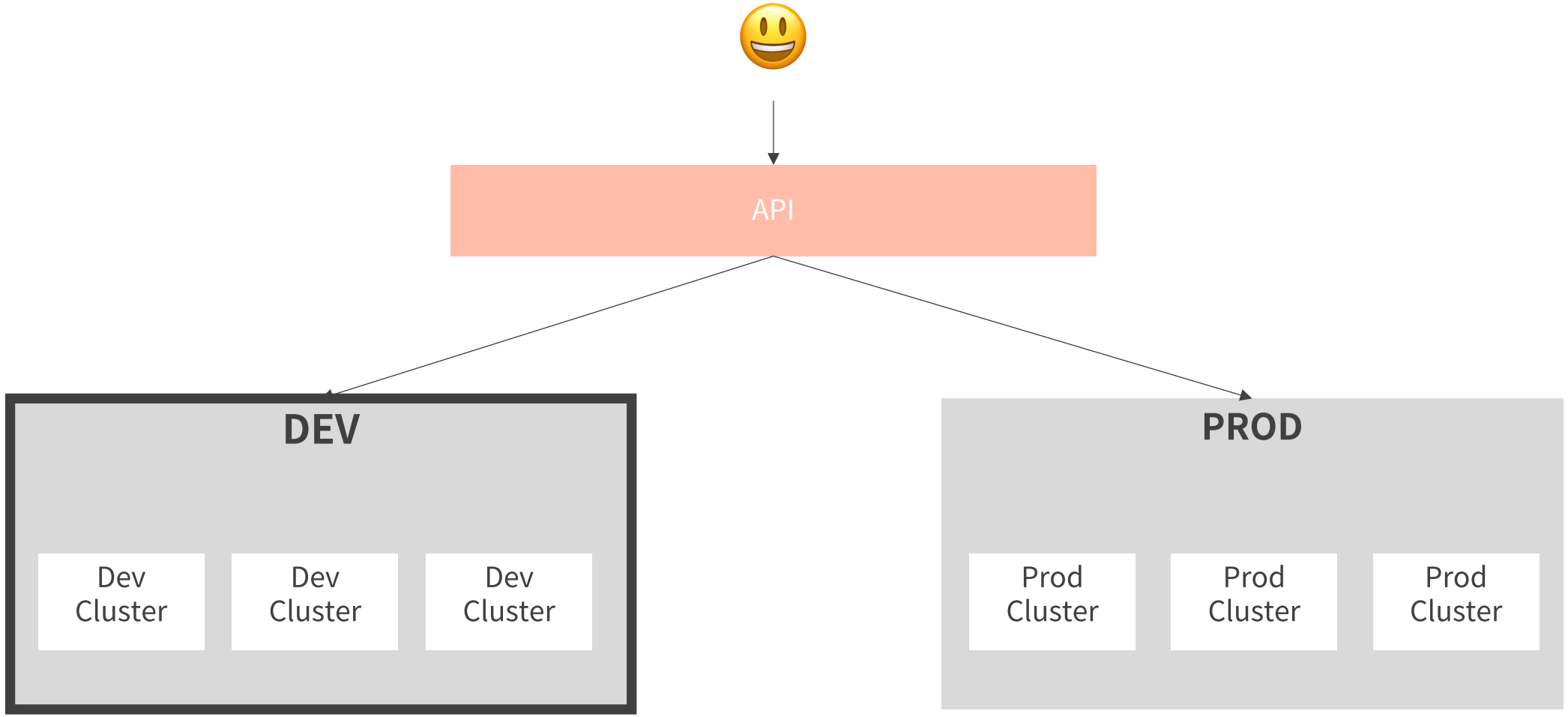
Build Firm Foundation

Resource별 QoS 정책 적용

SLA에 최소 보장 Performance 명시

Performance Baseline을 올리기 위한 연구 개발

Kubernetes Cluster As A Service



ACL 정책 적용

Kubernetes Multitanency

One Big Cluster

Control Plane을 공유

자원의 효율적 사용 가능

User / Org 별 namespace로 구분

모든 사용자가 인프라 자원 공유

다른 사용자의 Pod와 보안 격리가 어려움

Multi Cluster

사용자 개별 클러스터 생성

클러스터 단위 자원 추가

사용자 인증 단위로 클러스터 구분

클러스터 단위 인프라 자원이 구분됨

클러스터 단위 보안 정책 적용 가능

Obstacle: Multi-Cluster Management

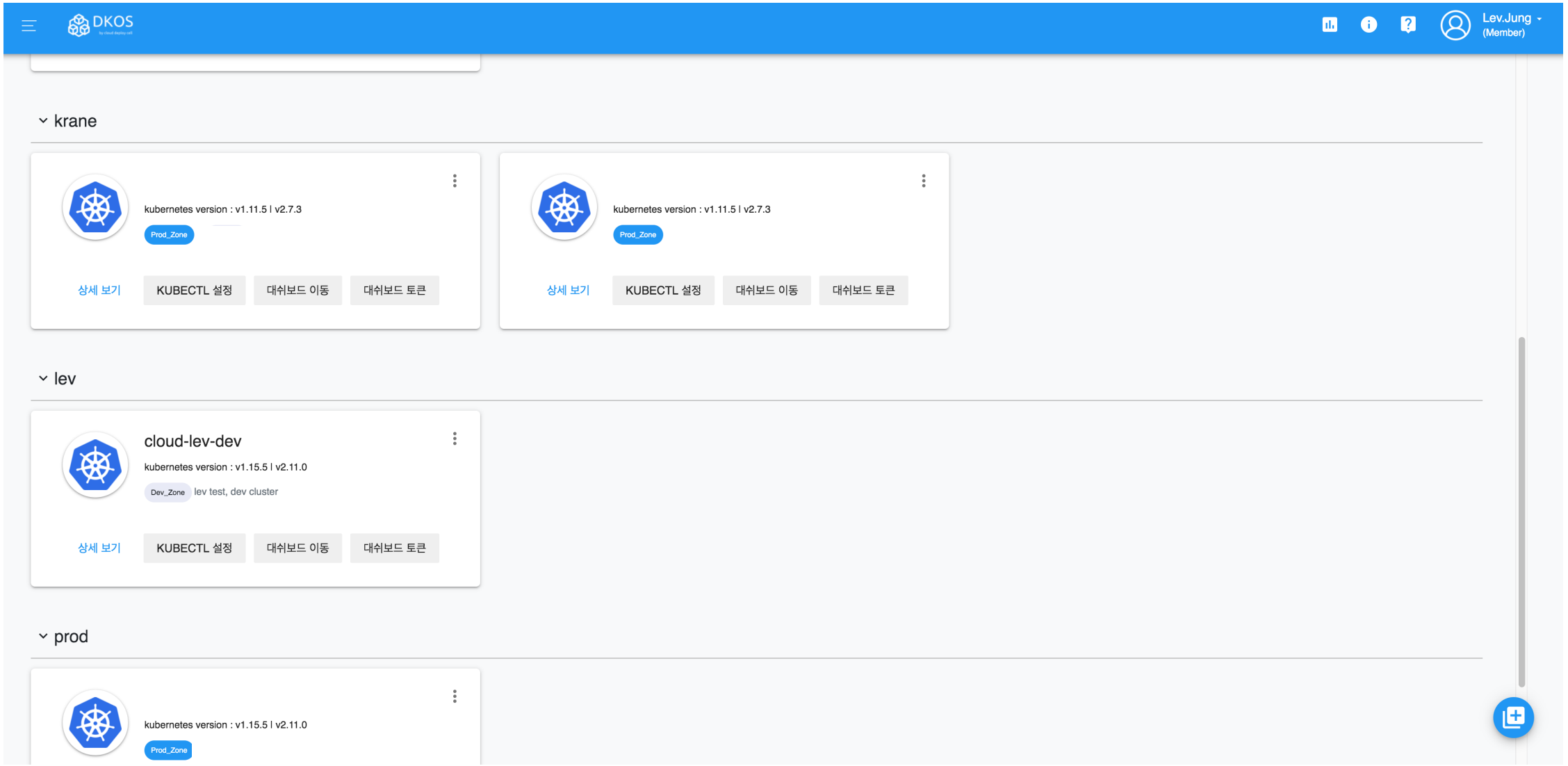
Web

사내 계정 시스템과 연동
참여 중인 클러스터만 표시
Kubernetes Token 기반으로 클러스터 Dashboard 접근

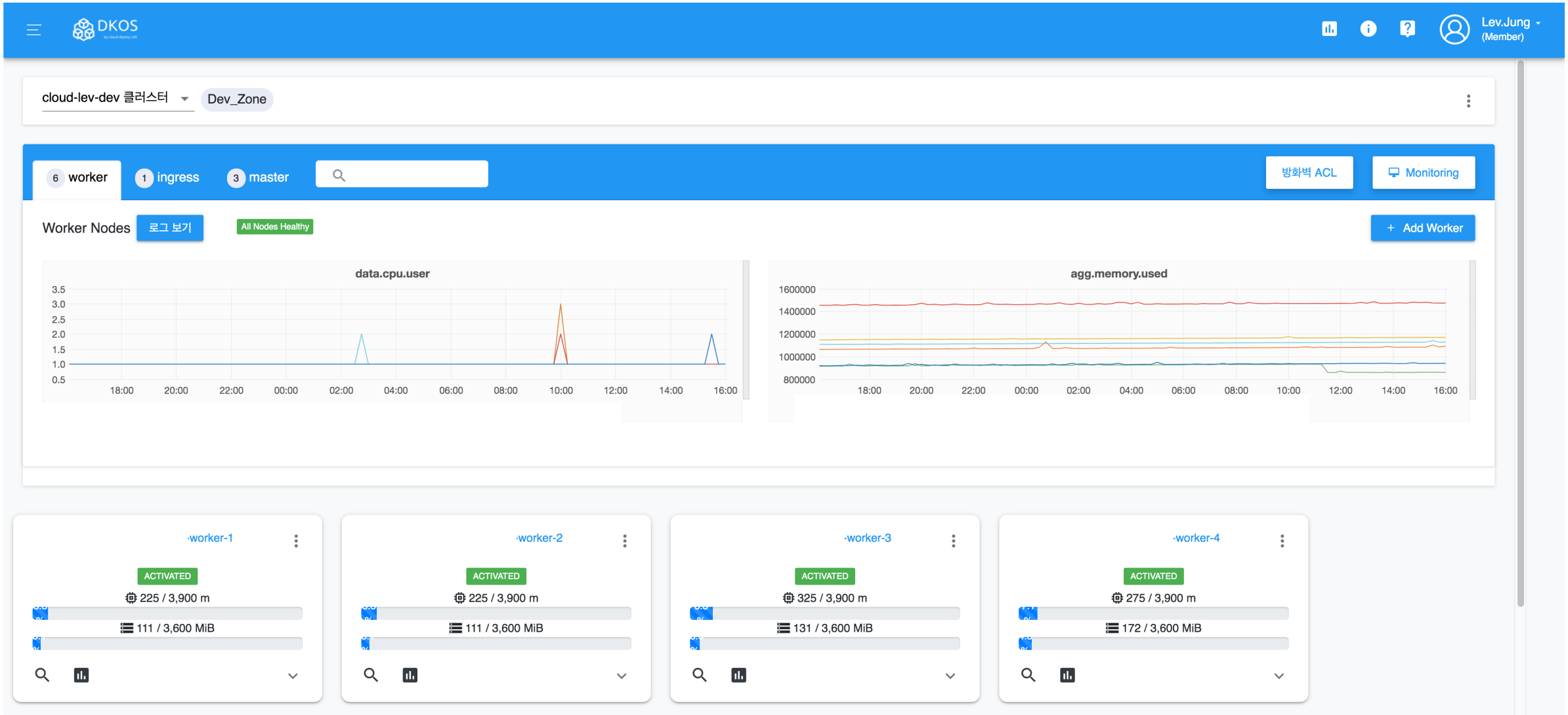
CLI

사내 자체 구축 Tool - dkosctl
인증 시스템 연동
참여중인 프로젝트를 CLI 환경에서 context 전환

DKOSv3 Multi-Cluster View



DKOSv3 Cluster Detail View



dkosctl Multi-Cluster management

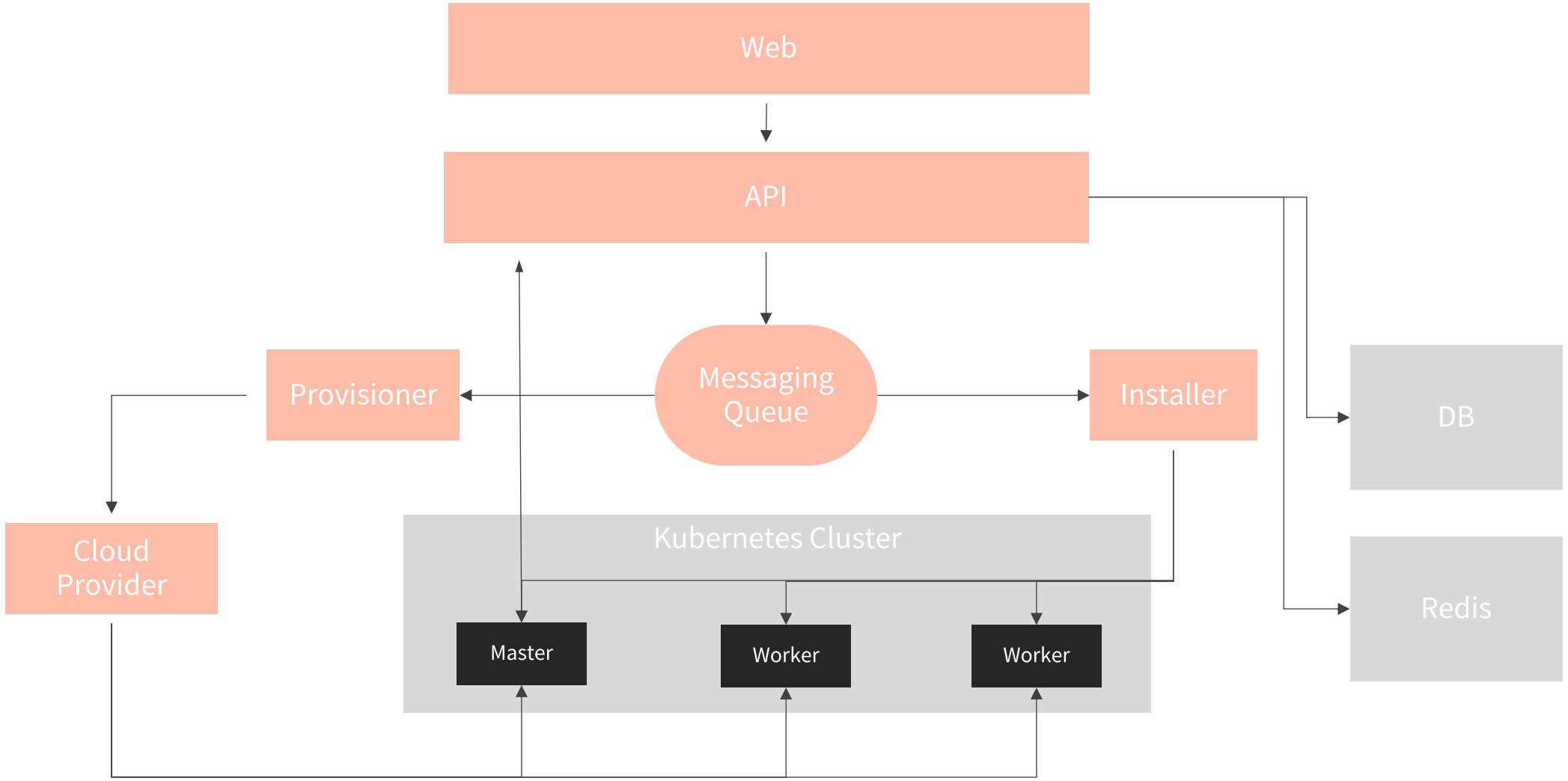
```
~
❯ (⎈ |krane-metric-context:default)dkosctl

DKOS controller
start with, "dkosctl configure"

To see help text, you can run:
  dkosctl [command] --help

Environment:
- $DKOSCTL_LDAP_ID # bypass login prompt
- $DKOSCTL_LDAP_PW
- $DKOSCTL_DKOSV3_HOST #  devel

Usage:
  dkosctl [command]

Available Commands:
  configure   configure
  console     openstack console
  create      create
  dashboard   open kubernetes dashboard
  delete      delete
  get         get
  help        Help about any command
  ns          namespace
  openstack   generate openstack env(osenv)
  project     openstack project
  quota       openstack quota
  server      openstack server
  ssh         ssh
  version     version
  volume      openstack volume

Flags:
      --debug   debug mode
  -h, --help    help for dkosctl

Use "dkosctl [command] --help" for more information about a command.
```

dkoctl cli 제공 기능

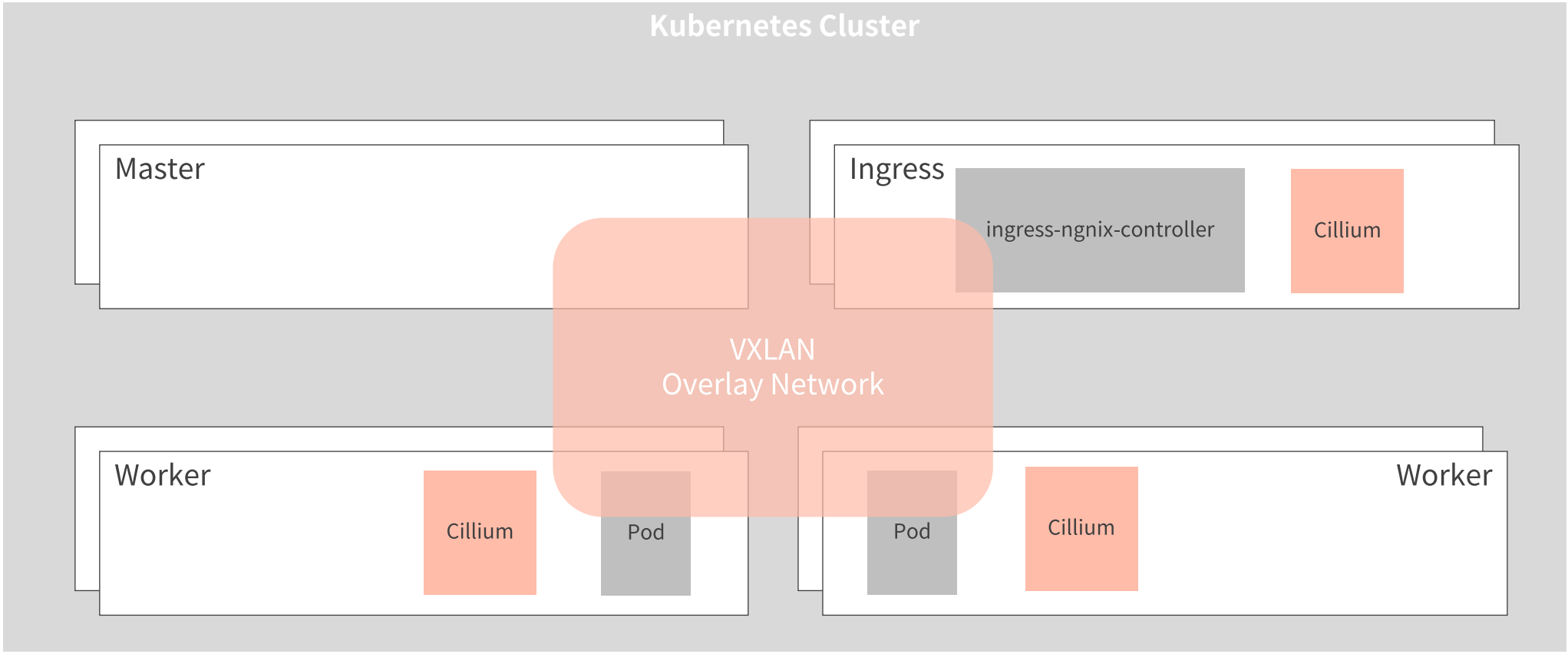
```
                          | v1.15.11   | v2.11.1   |                          | true      |
                          | v1.15.5    | v2.11.0   |                          | false     |
   4910 | cloud-lev-dev     | v1.15.5    | v2.11.0   |                          | false     | lev test, dev c
                          | v1.11.5    | v2.7.3    |                          | true      |
                          | v1.11.5    | v2.7.3    |                          | true      |
>                         | v1.11.5    | v2.7.3    |                          | true      |
  -------+-----------------+------------+-----------+--------------------------+-----------+-----------------
    ID   |       NAME      | KUBERNETES | KUBESPRAY | IMS(ROOT) |     CREATED      | OPENSTACK |   DESCRIPTION
> < 6/6
```

dkosctl multicluster 선택 예시

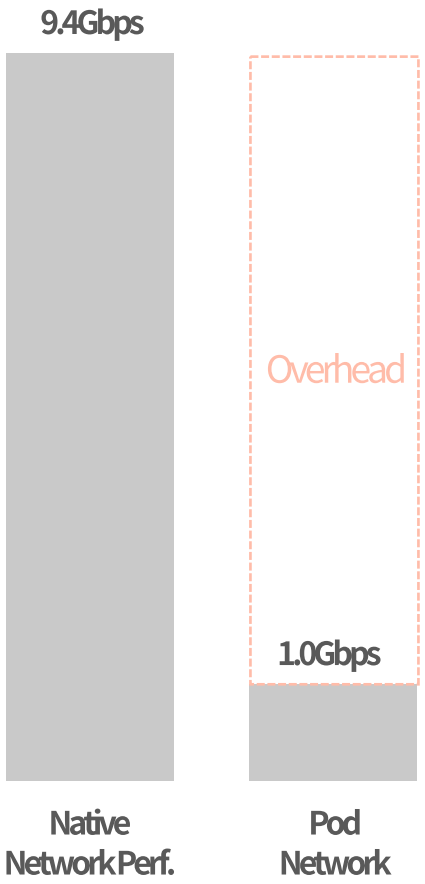
DKOSv3 Architecture Overview



DKOSv3 Cluster



Obstacle: Vxlan Performance



Tunnel Overhead

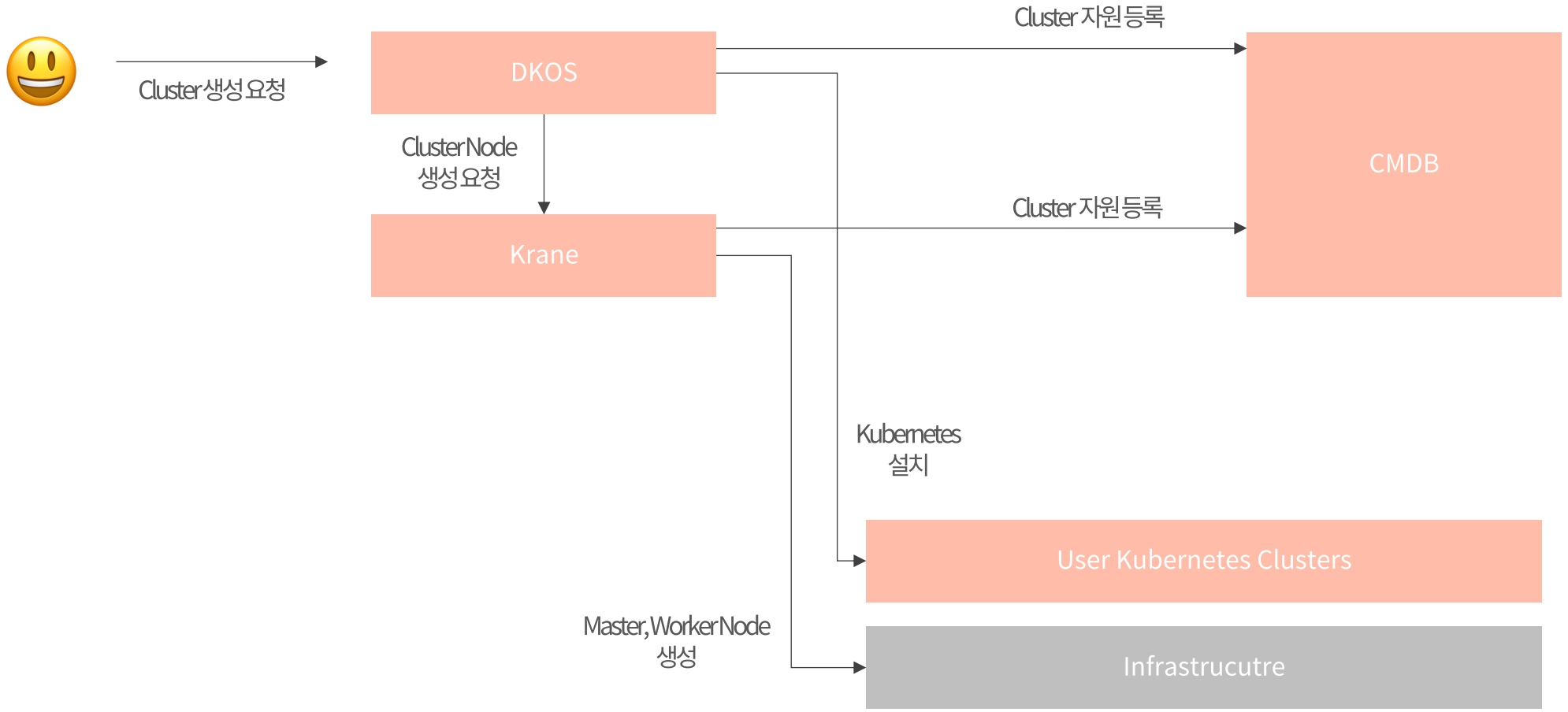
Cillium Datapath 구조 분석

VxLAN encapsulation / decapsulation에 의한
오버헤드 발생 인지

Pod간 Overlay Network 필요하지 않는 경우 HostNetworkMode

SR-IOV와 같이 NIC Offloading 기능 활용 개발 방향 수립

Code to App: Cluster



Obstacle: Infra Resources

Image Based Infra

Master, Worker, Ingress 모두 Krane을 기반으로 생성하는 VM

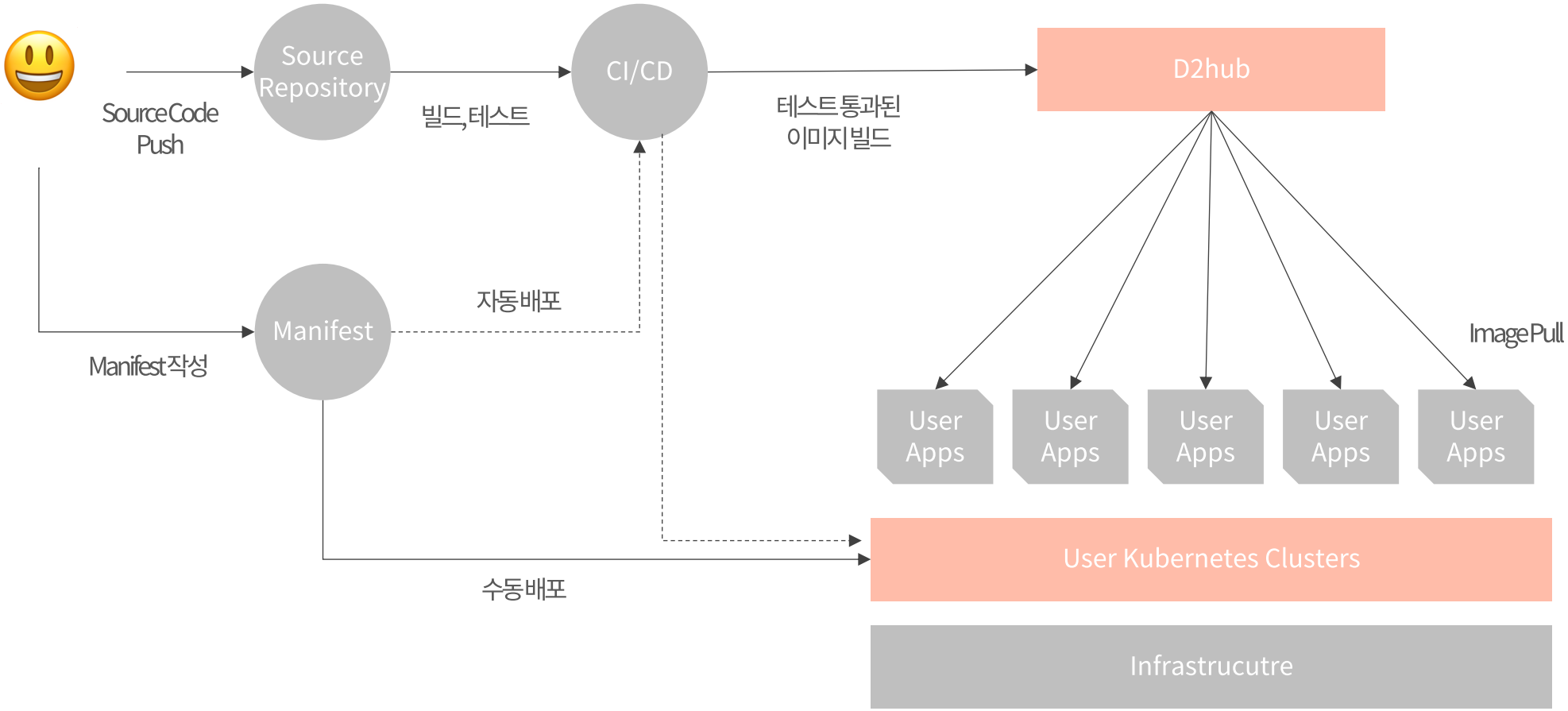
VM 이미지는 인프라팀에서 사내 보안 규정을 만족하는 이미지(Base Image) 생성

Base Image 기반으로 매주 자동으로 빌드

노드 내부 설정 오염 또는 소프트웨어 이슈 해결은 Rebuild or Re-Create

보안 패치, 커널 업그레이드가 필요한 경우도 Rebuild or Re-Create

Code to App: Deployment



Obstacle: Container Image

D2hub

사내 Github Enterprise와 연동하여 Repo의 Branch, Tag 기반으로 Docker Image 빌드 기능 지원

LDAP과 연동하여 인증과 권한 관리

사내 보안 도구와 연계하여 Image 내 알려진 취약성 검사 가능

인가된 클러스터에 이미지를 자동 배포 기능 제공

Obstacle: Manifest

K2hub

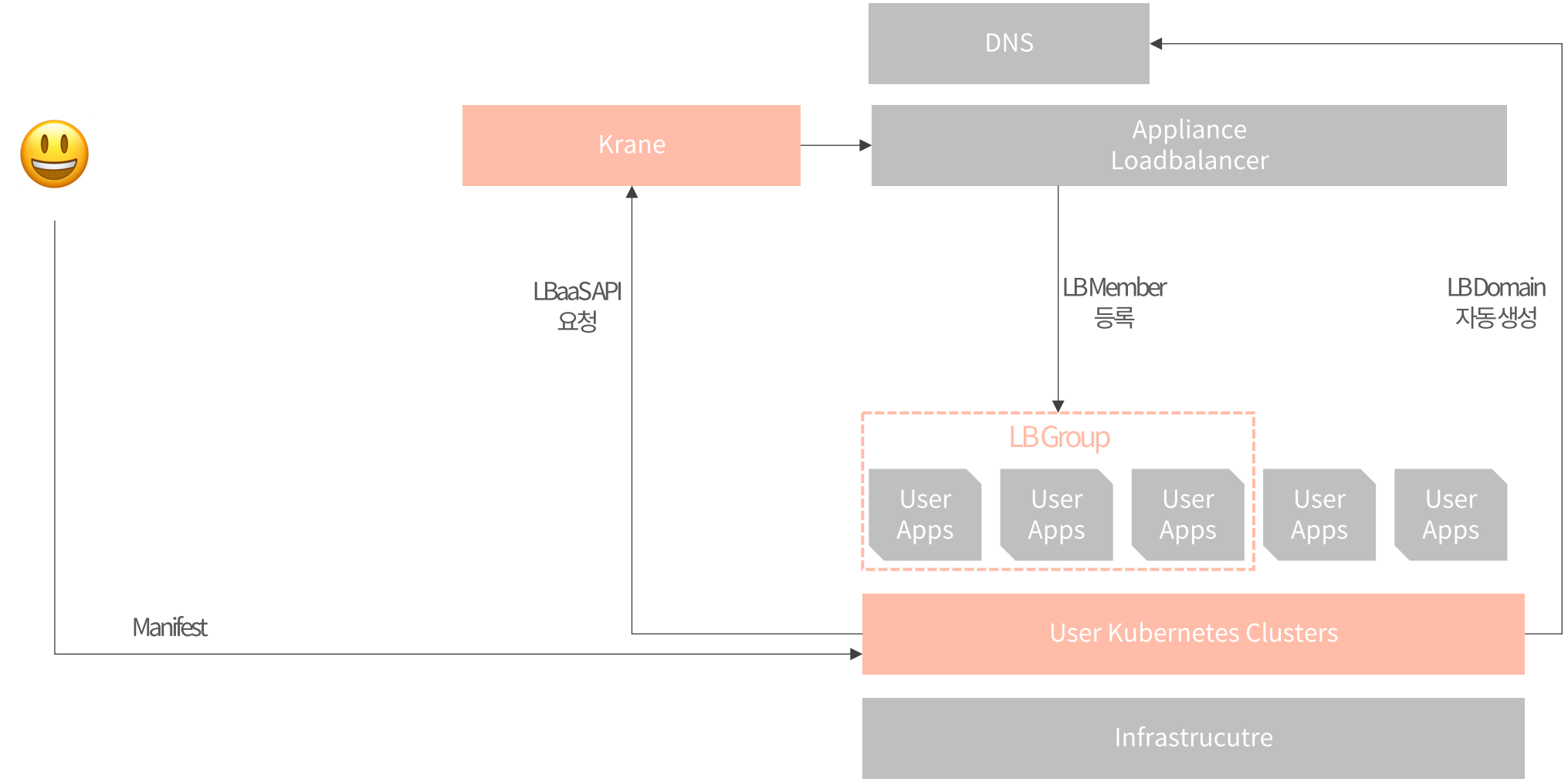
Kakao Helm Chart museum

사내에서 관리하는 Application / Platform의 Chart 사용 가능

잘 알려진 외부 Chart로 제공

Web 기반으로 인가된 클러스터에 Chart 배포 기능 제공

Code to App: Connect to LoadBalancer



Obstacle: DSR

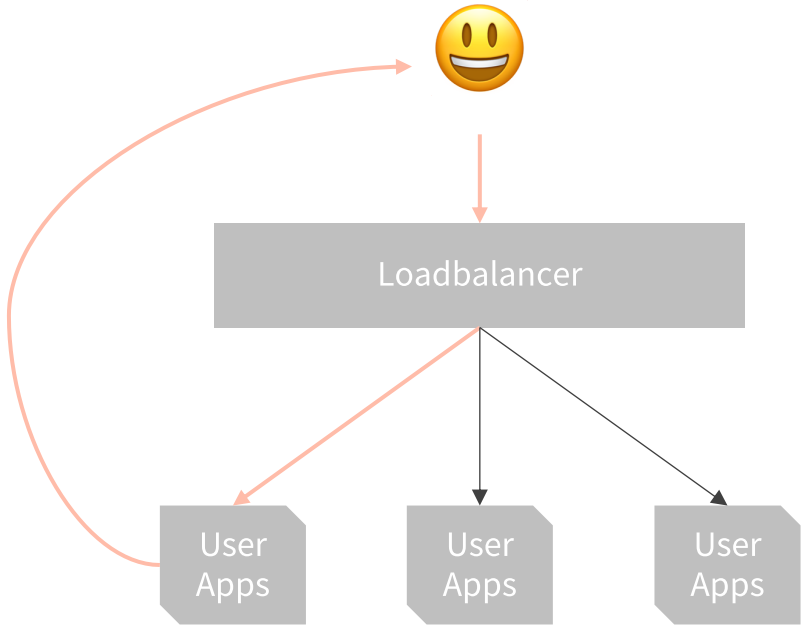
Direct Server Return(DSR)

카카오 사내 LB 기본 정책

LB와 Member간의 IP-in-IP 터널이 필요

DSR-Manger 구현

자동으로 Ingress Node, Worker Node에서 DSR 설정 역할



Obstacle: LB Controller

Custom LB Controller

OpenStack이 관리하지 않는 PM의 경우도 LB Member로 참여 가능

Node의 Label 기반으로 LB Member Selection

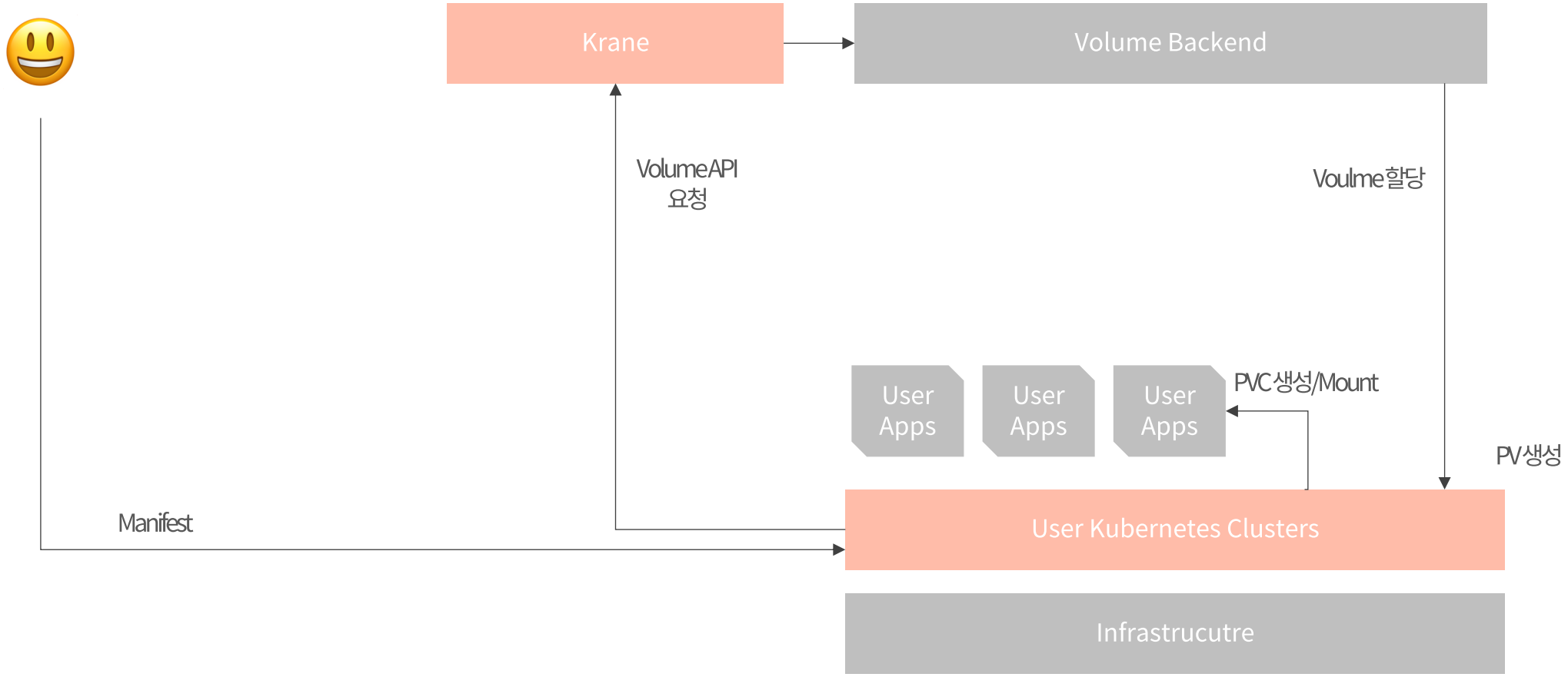
Dev / Prod Cluster 별로 다른 LB Provider 선택 가능

Obstacle: Domain

inhouse-dns-controller

Ingress 설정에 특정 형식으로 Host 설정을 한 경우 자동으로 DNS에 domain 등록

Code to App: Connect to Volume



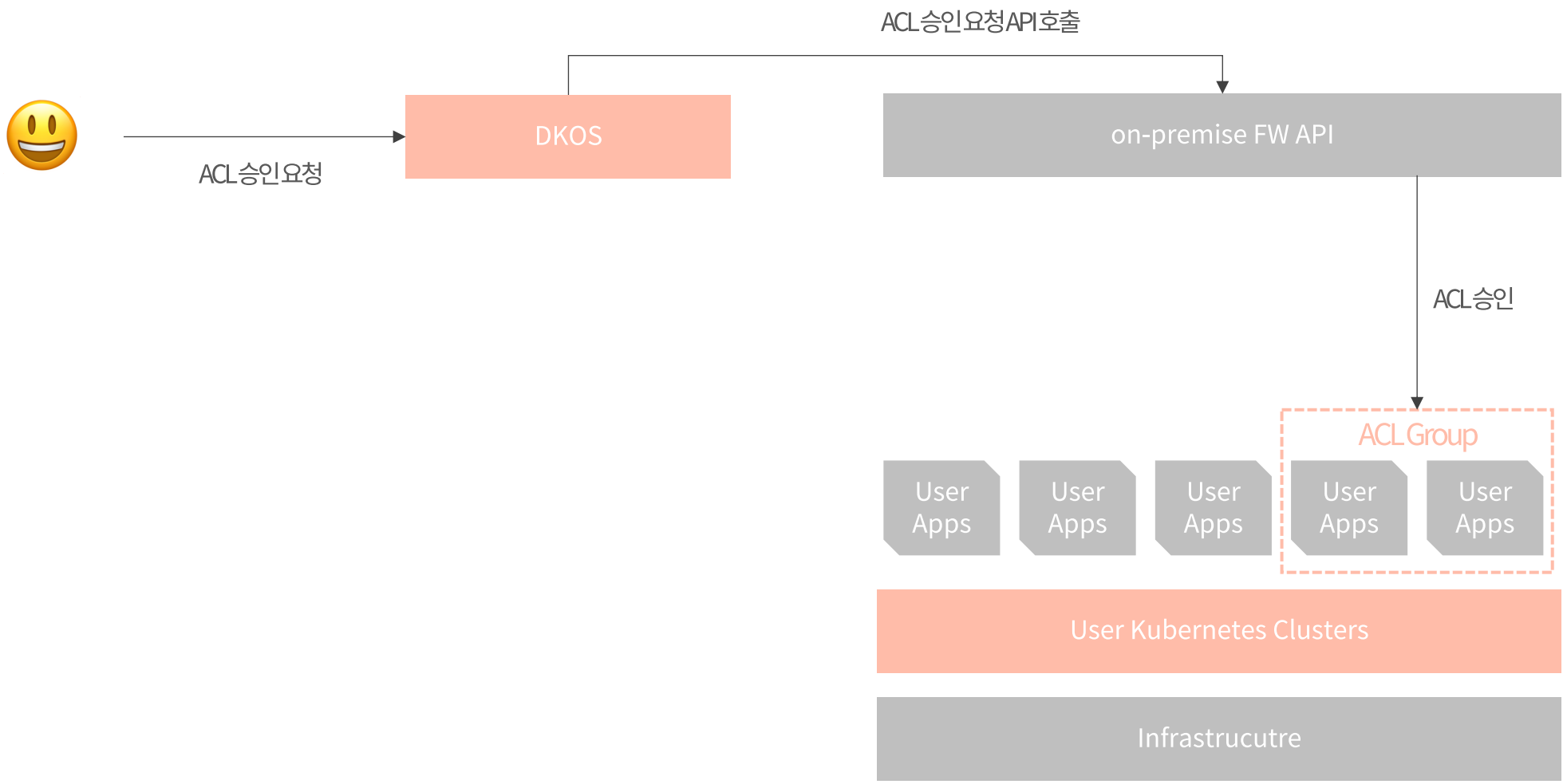
Obstacle: Cinder Controller

CSI Cinder

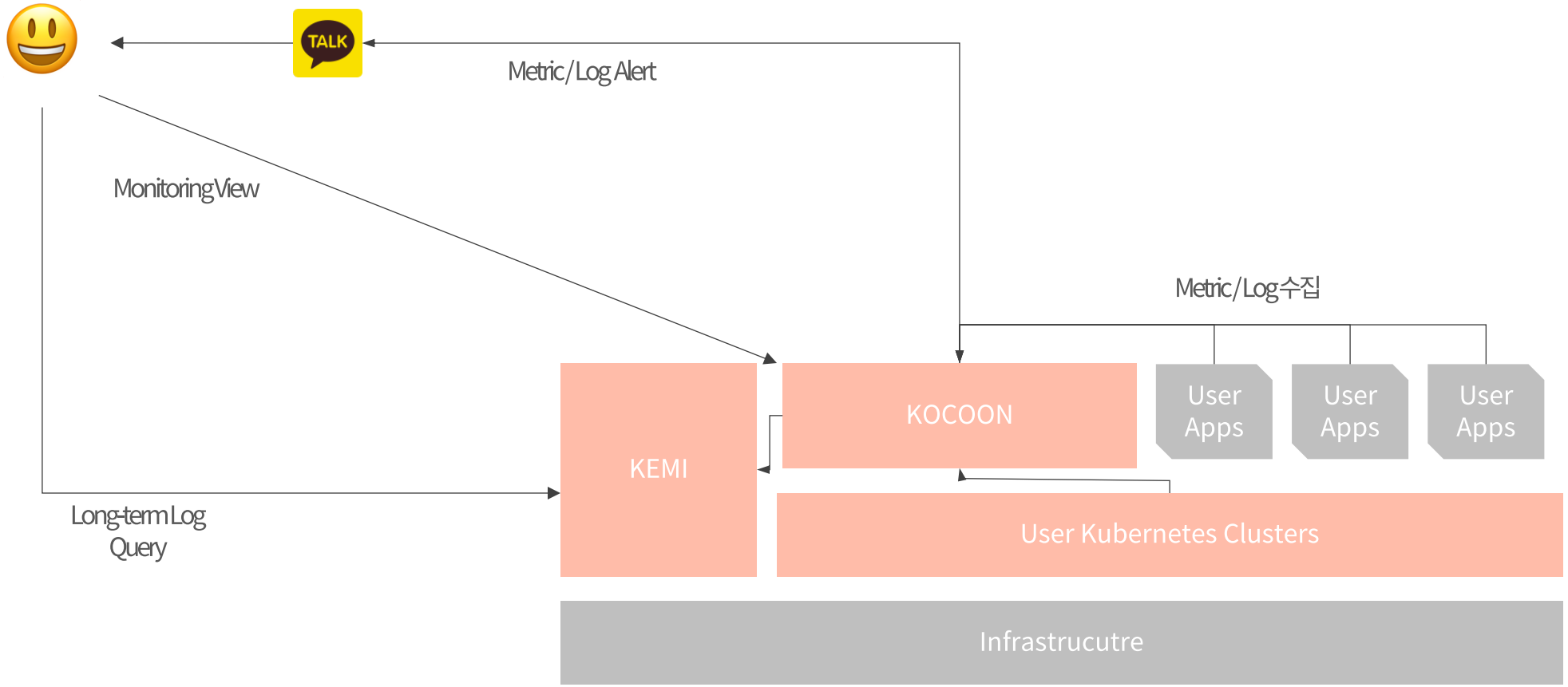
Cinder Controller를 사용하기 위해 Kubernetes Cloud Provider가 OpenStack API에 주는 부하가 상당

부하를 줄이기 위해 CSI(Container Storage Interface)를 활용한 새로운 Storage Controller를 구현

Code to App: Connect to ACL



Code to App: Monitoring



Obstacle: Resource Monitoring

KOCOON

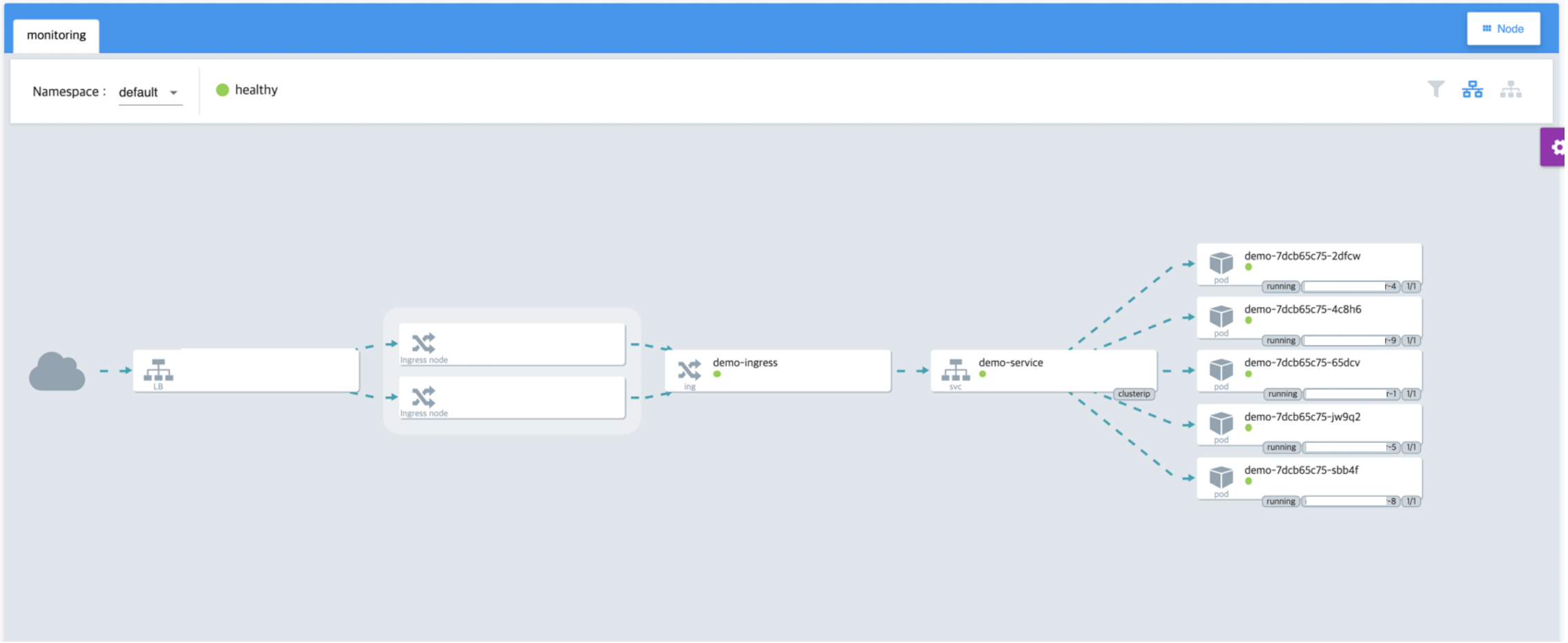
서비스 클러스터 별로 모니터링 리소스를 분리

Prometheus 기반의 메트릭 수집

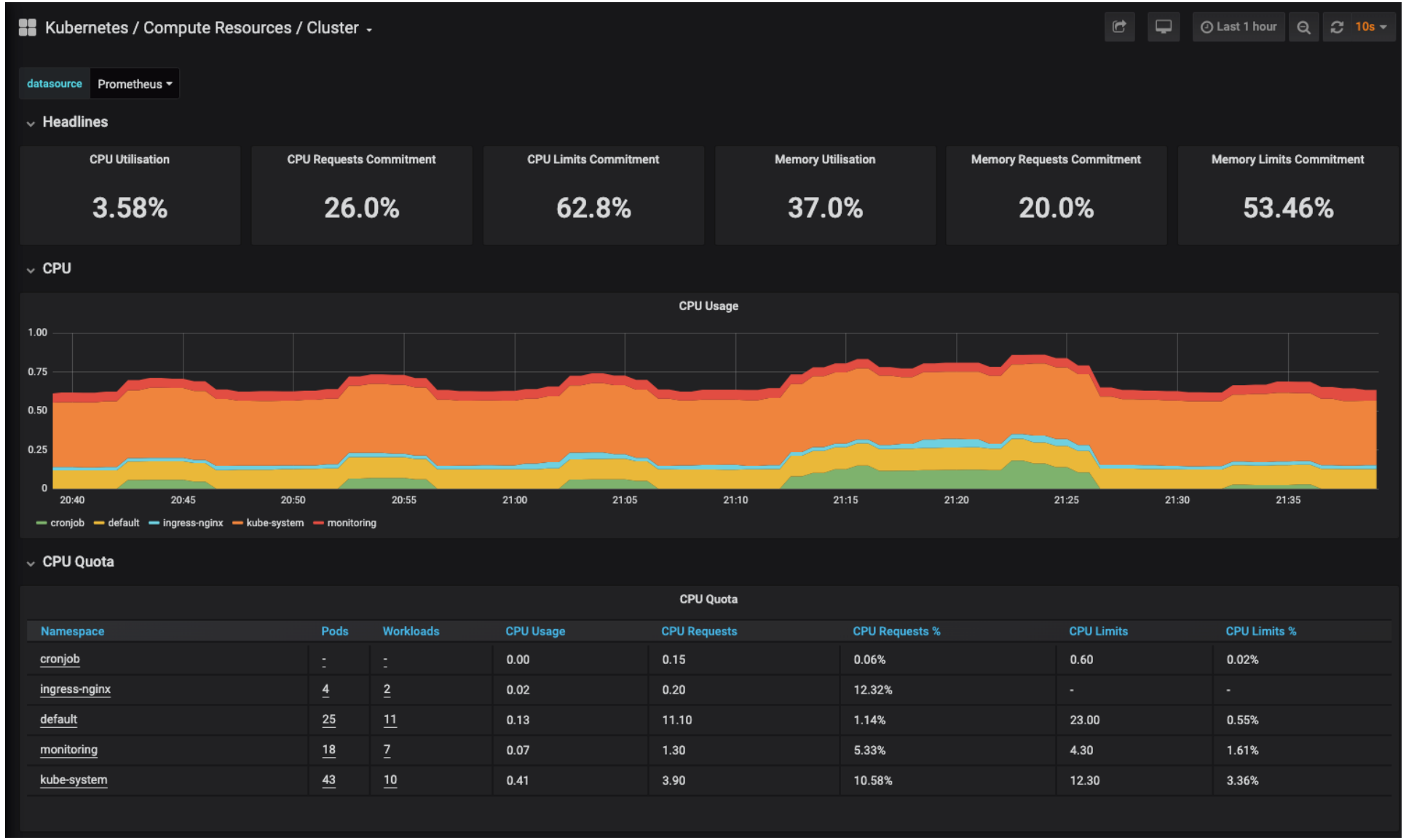
KOCOON-Hermes(Fluentd 기반)으로 Log Routing

KOCOON-CUPIDO로 Alert을 카톡 전송

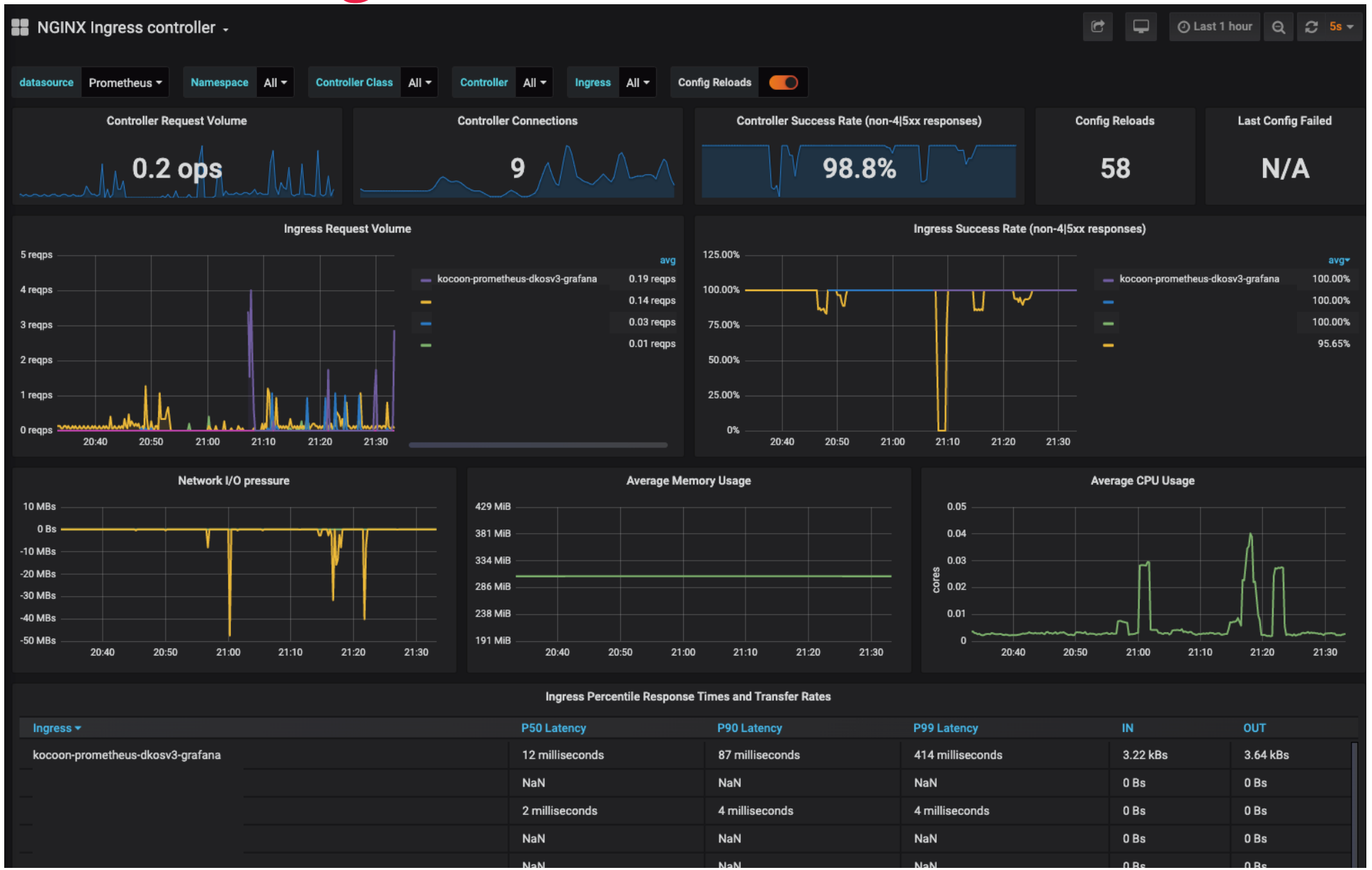
DKOSv3 Monitoring View



DKOSv3 Monitoring View



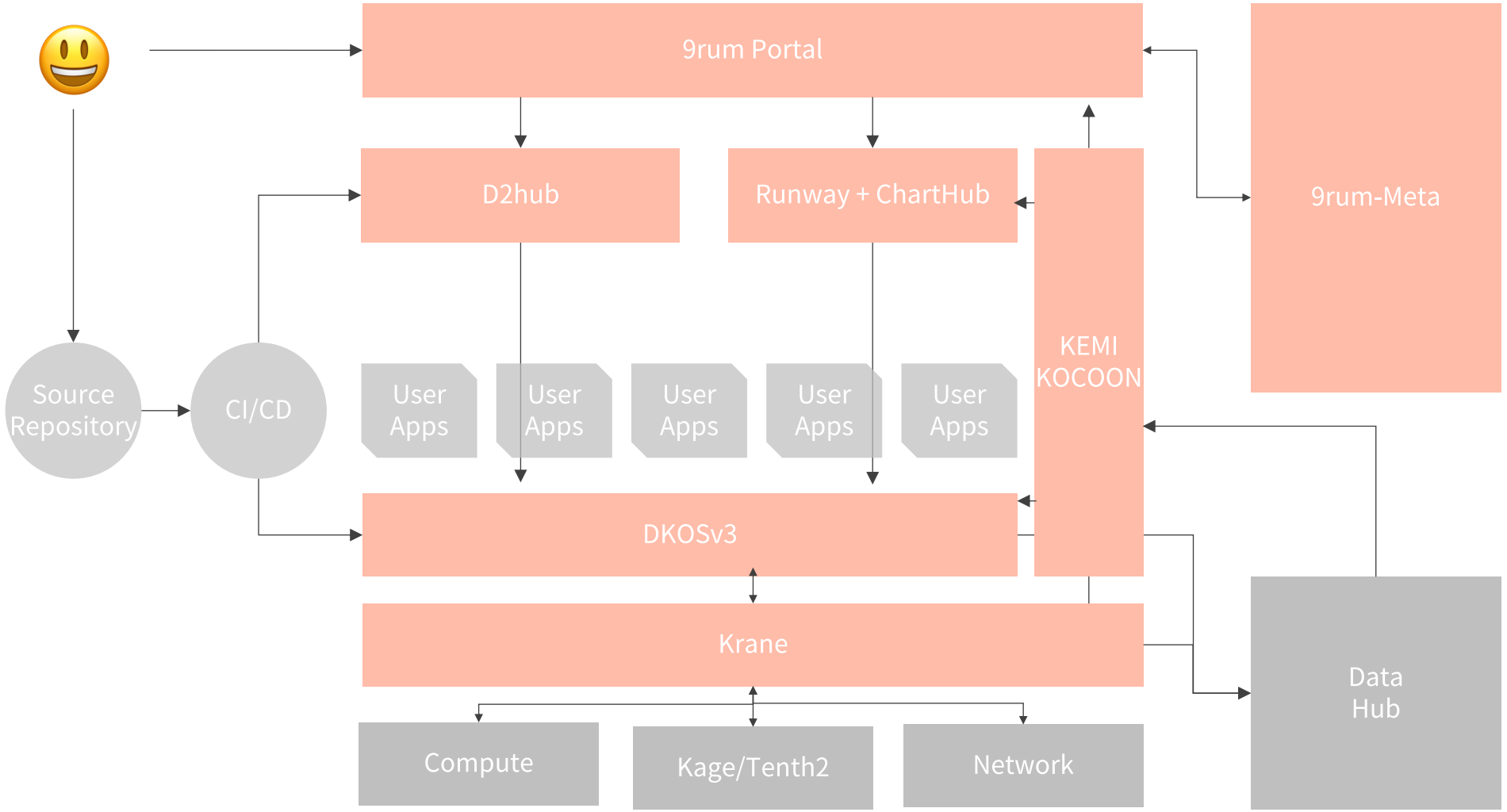
DKOSv3 Monitoring View



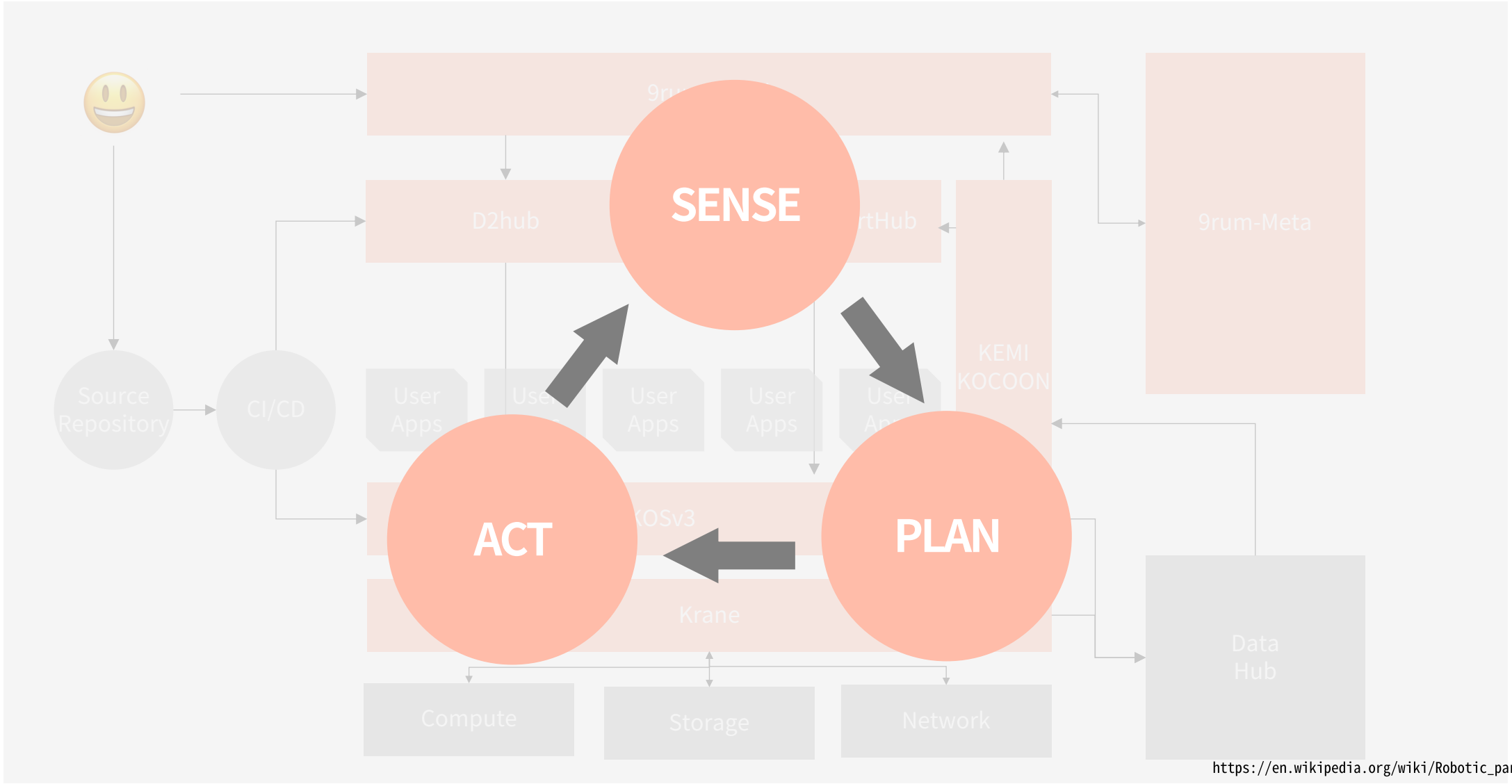
DKOSv3 Monitoring View

04 Summary

Again, Cloud Native Technologies at Kakao



Toward to Fully Automated Cloud



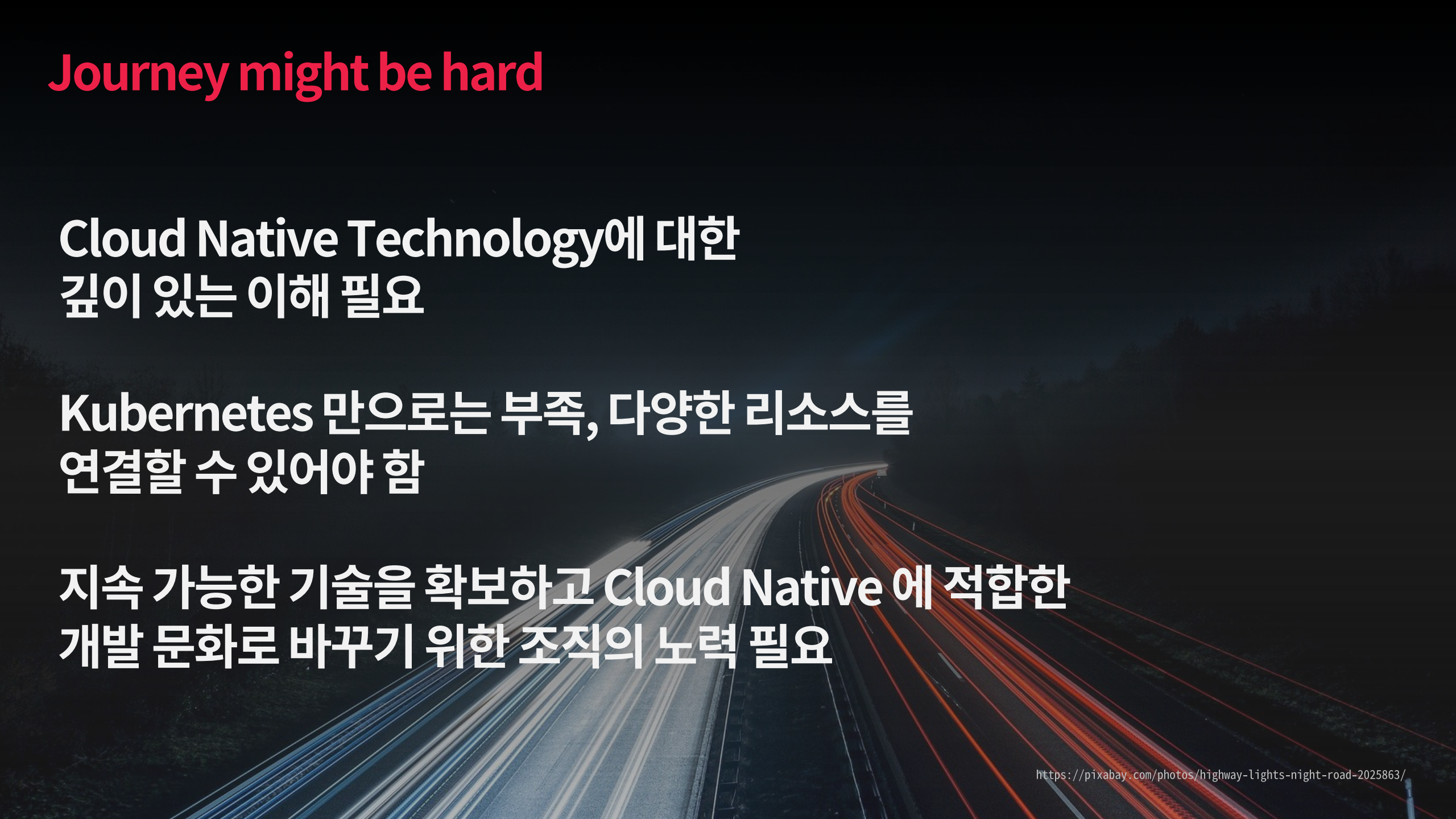
https://en.wikipedia.org/wiki/Robotic_paradigm

Journey might be hard

Cloud Native Technology에 대한
깊이 있는 이해 필요

Kubernetes 만으로는 부족, 다양한 리소스를
연결할 수 있어야 함

지속 가능한 기술을 확보하고 Cloud Native 에 적합한
개발 문화로 바꾸기 위한 조직의 노력 필요



https://pixabay.com/photos/highway-lights-night-road-2025863/

질문과 답변

감사합니다

we are hiring!