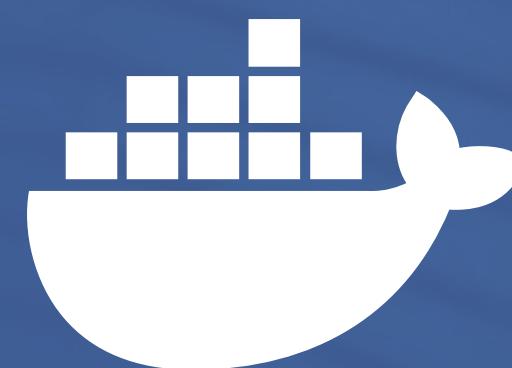


초보를 위한 쿠버네티스 안내서

쿠버네티스 시작하기

KIM CHUNG SUB
Twitter:  GitHub:  subicura



쿠버네티스 시작하기

컨테이너 오케스트레이션

왜 쿠버네티스인가?

어떤걸 배울까?

쿠버네티스 알아보기

쿠버네티스 소개

쿠버네티스 아키텍처

실습

쿠버네티스 실습하기

컨테이너 오케스트레이션

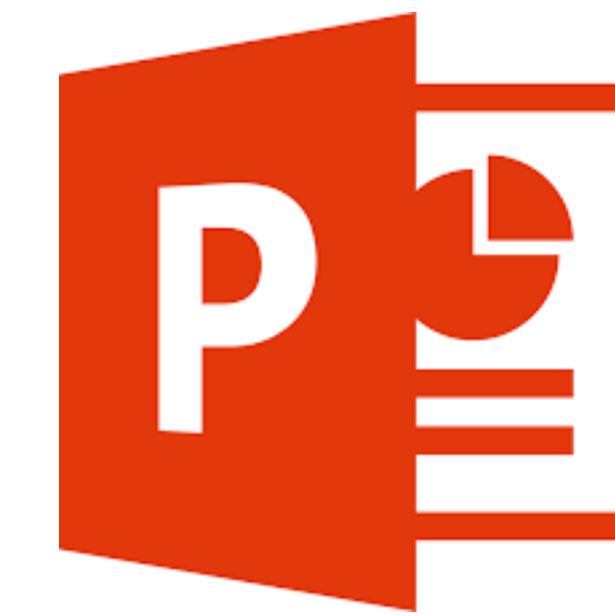


서버를 관리한다는 것

서버의 상태를 관리하기 위한 노력



서버를 관리한다는 것



A부터 Z까지
화면 하나 하나 잘 캡쳐하자..



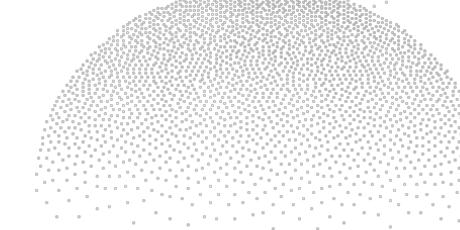
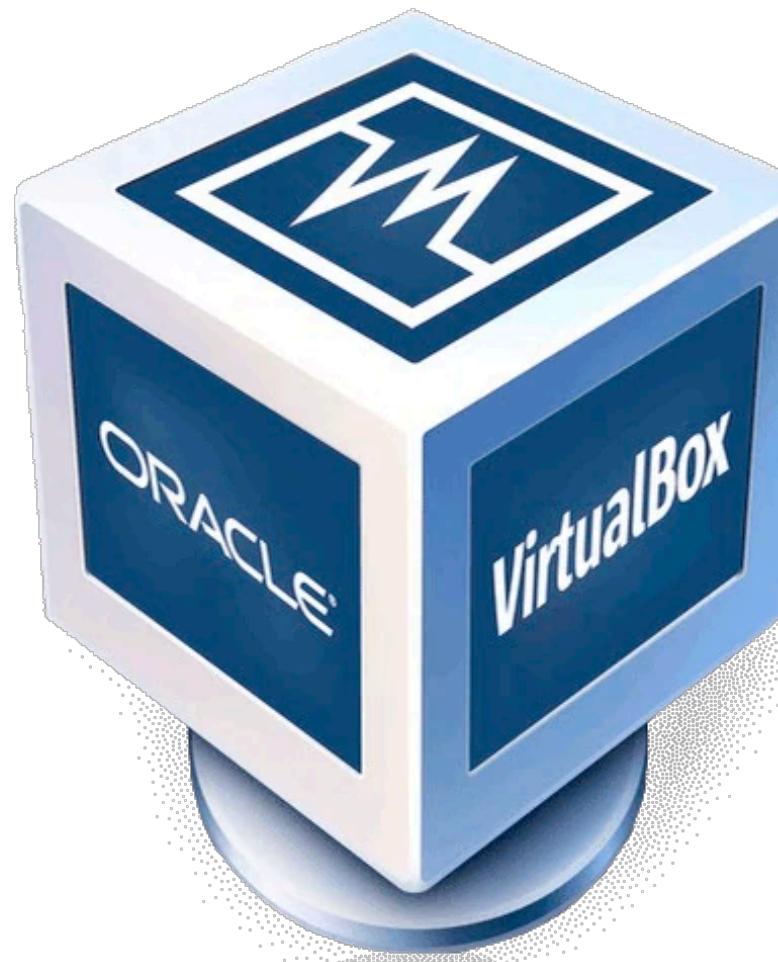
따라해도
잘 안돼..ㅠㅠ

서버를 관리한다는 것



문서보다는 코드지!
이제 스크립트 직접 입력 X

서버를 관리한다는 것

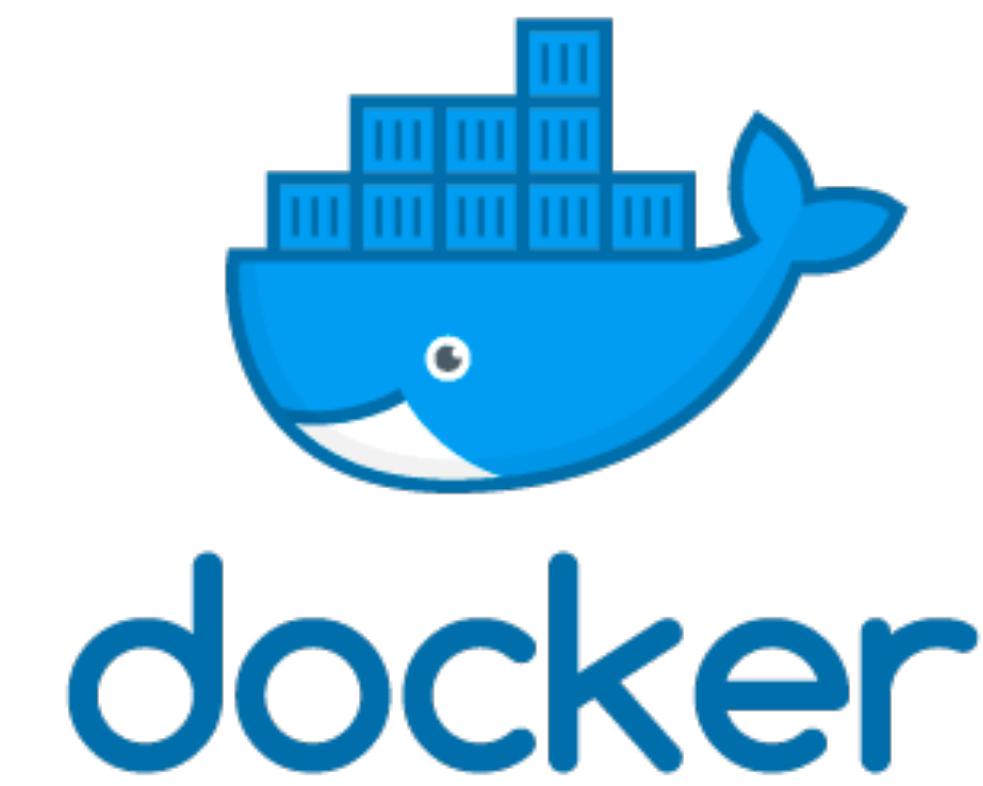


서버 하나에 가상화된 여러개!
초콜 느리고 관리가 불편하지만.. 좋은데?

다른 클라우드는?



도커의 등장



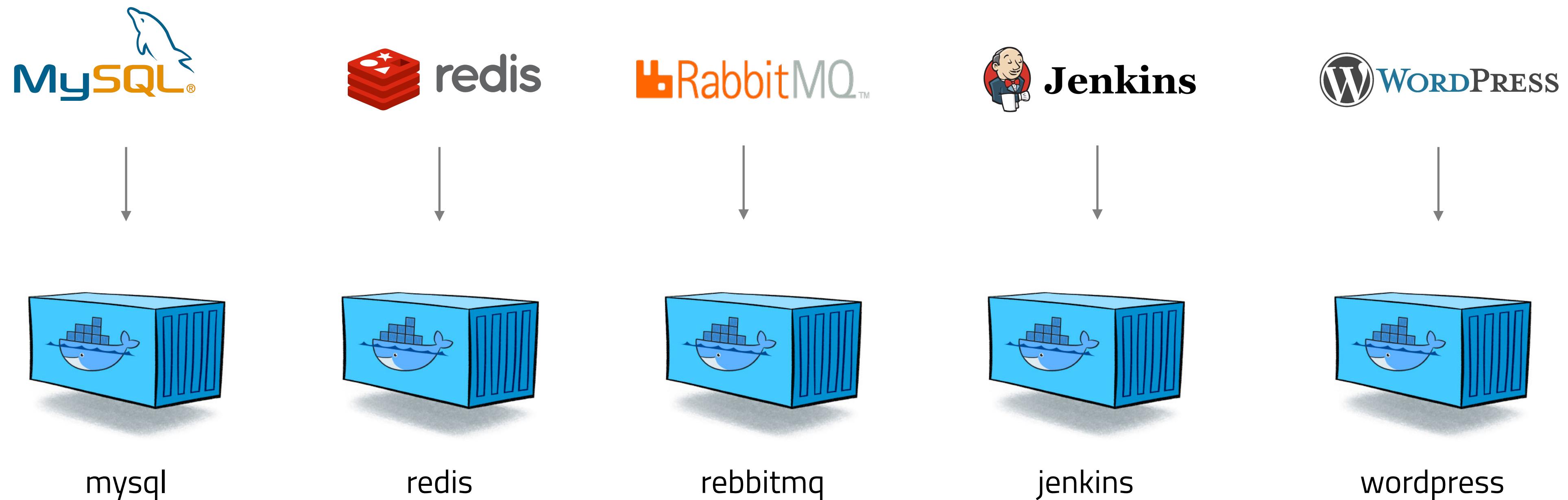
모든 실행환경을 컨테이너로!
어디서든 동작하고 쉽고 효율적이라고? ~~실화~~

도커 컨테이너

컨테이너의 특징

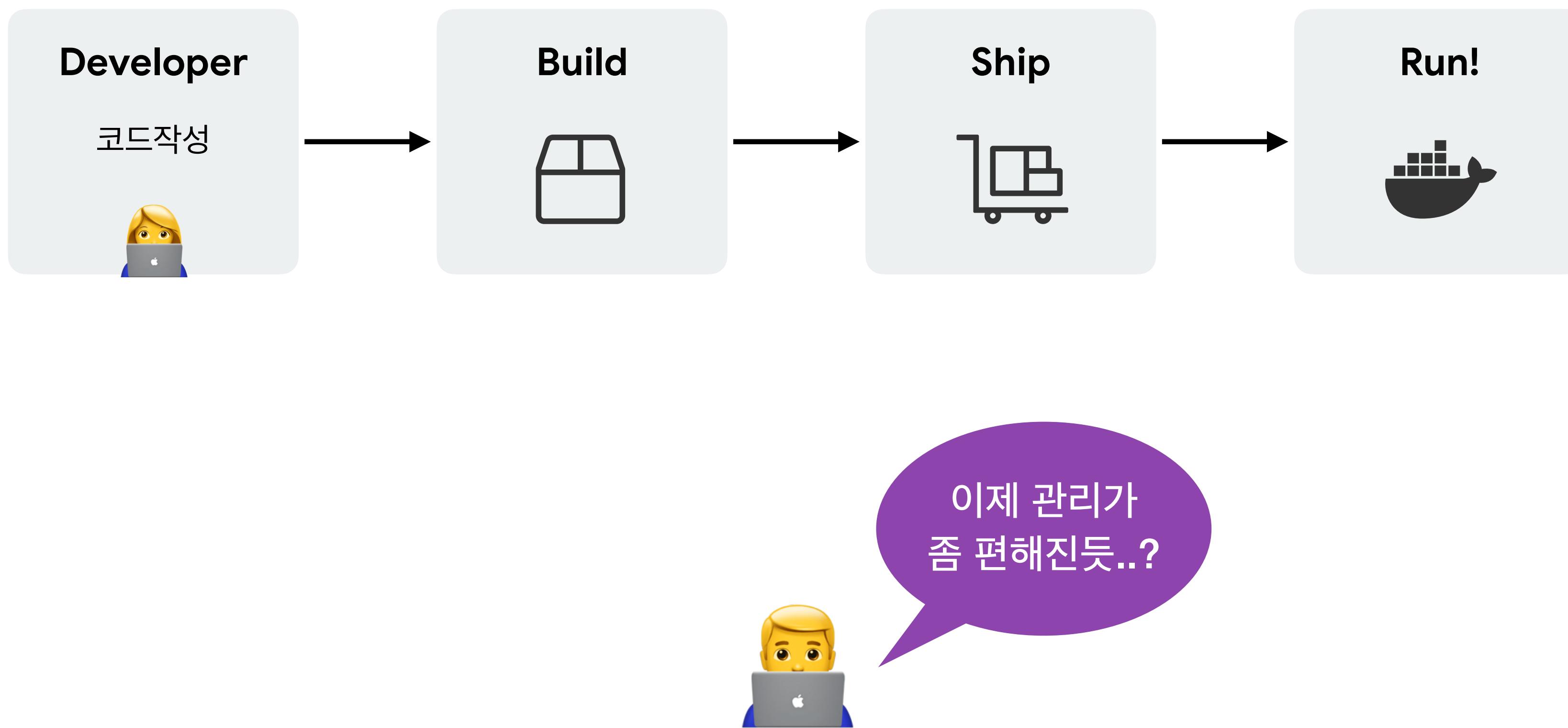
- 가상머신과 비교하여 컨테이너 생성이 쉽고 효율적
- 컨테이너 이미지를 이용한 배포와 롤백이 간단
- 언어나 프레임워크에 상관없이 애플리케이션을 동일한 방식으로 관리
- 개발, 테스팅, 운영 환경은 물론 로컬 피시와 클라우드까지 동일한 환경을 구축
- 특정 클라우드 벤더에 종속적이지 않음

도커의 등장



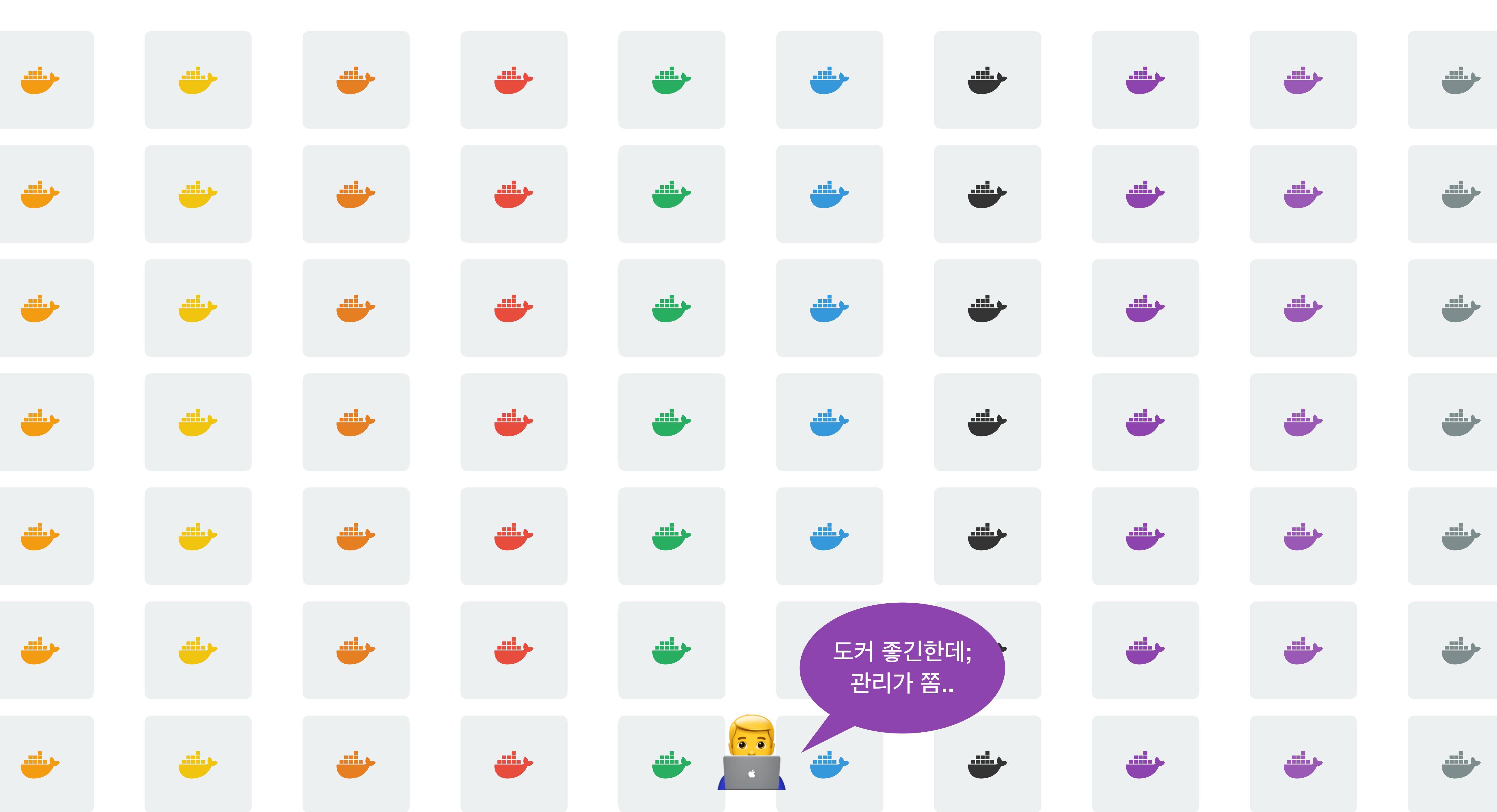
containerization!!

도커의 등장

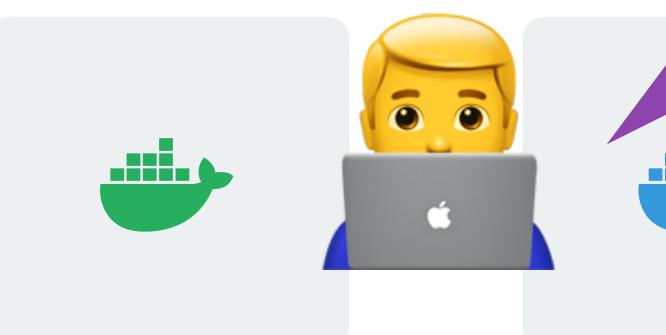


도커의 등장





도커 좋긴한데;
관리가 좀..



도커 그 이후



Deployment

1. 배포는 어떻게 할까?

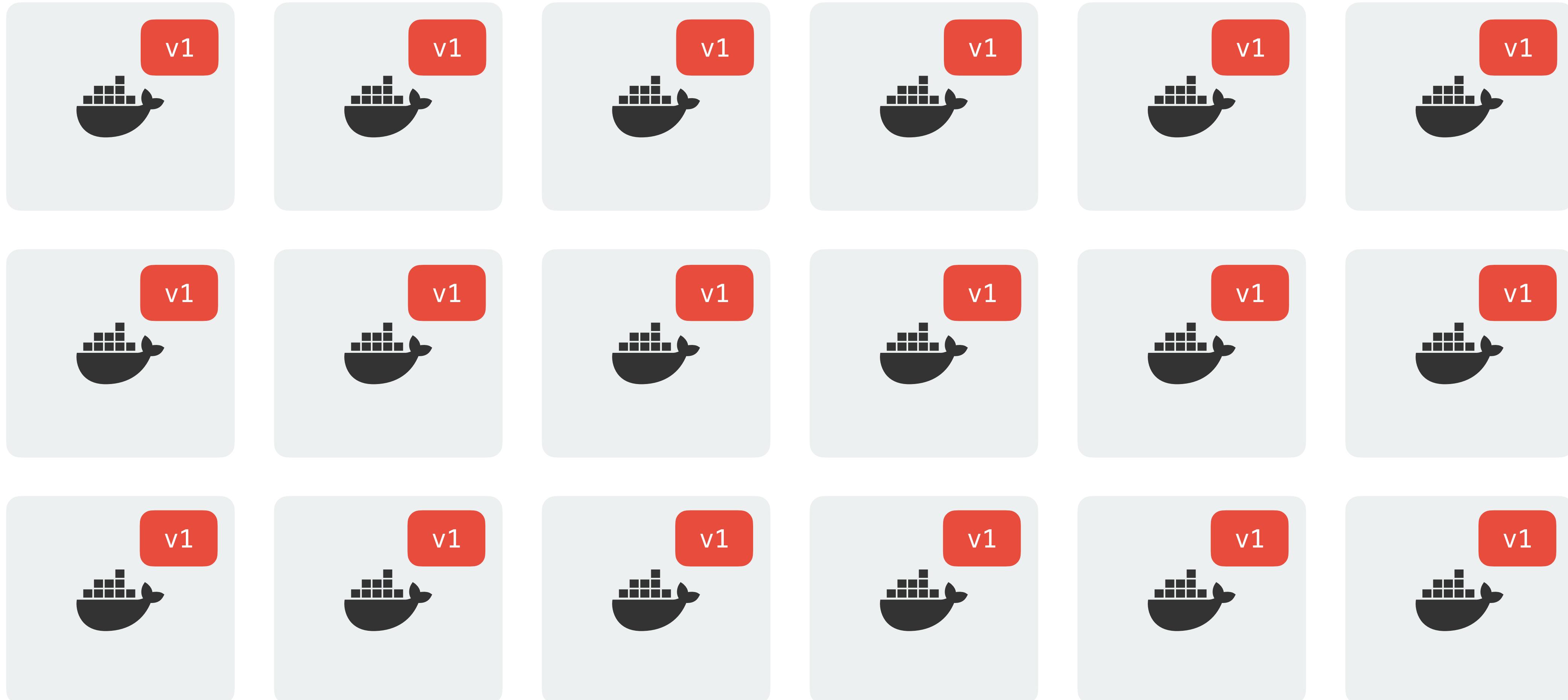
도커 그 이후



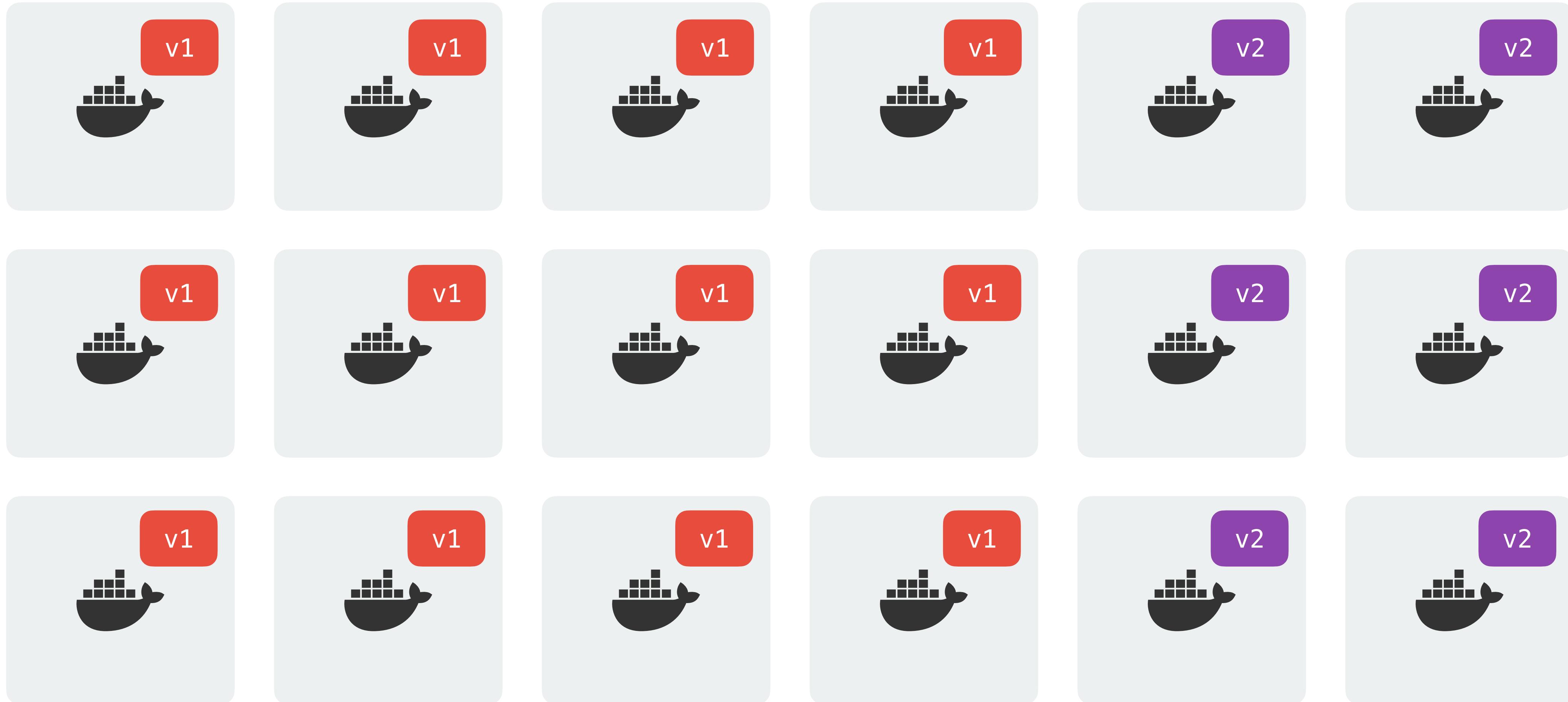
도커 그 이후



도커 그 이후



도커 그 이후



도커 그 이후



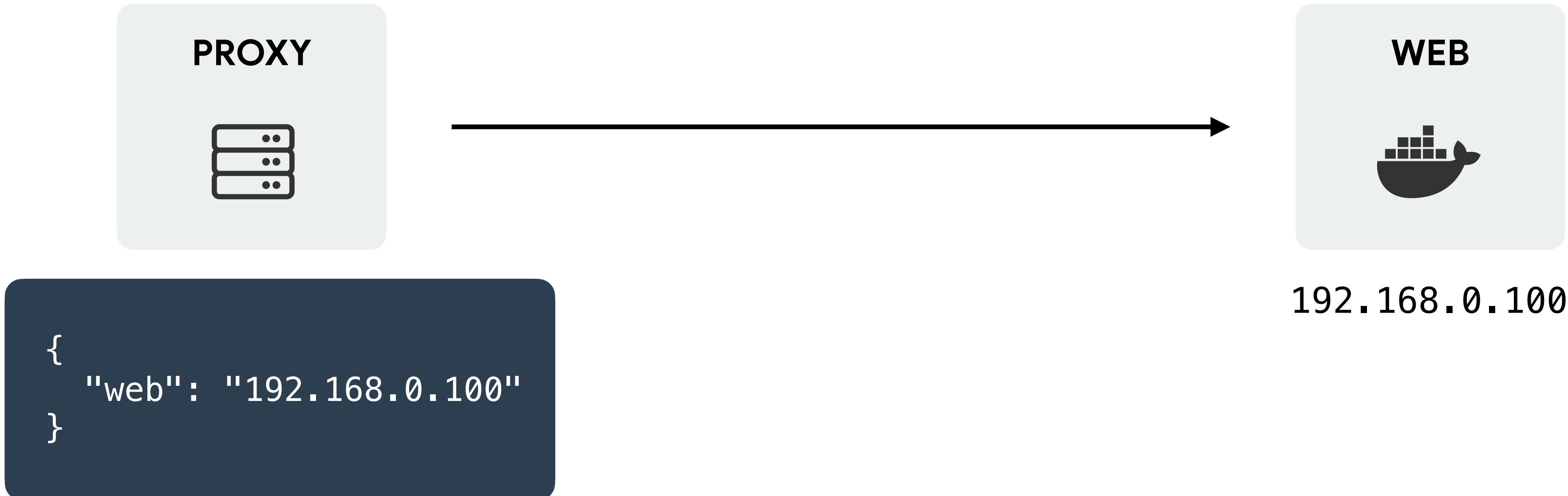
도커 그 이후



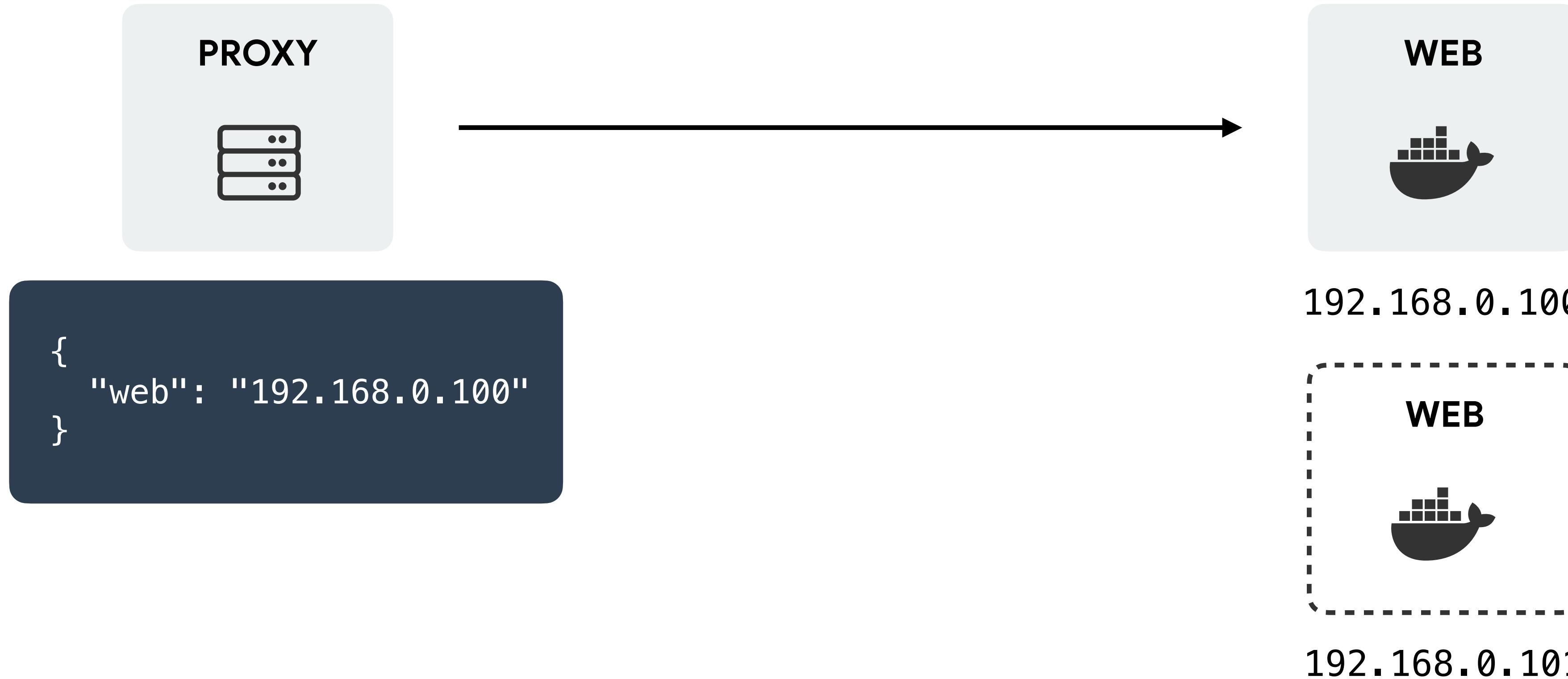
Service Discovery

2. 서비스 검색은 어떻게 할까?

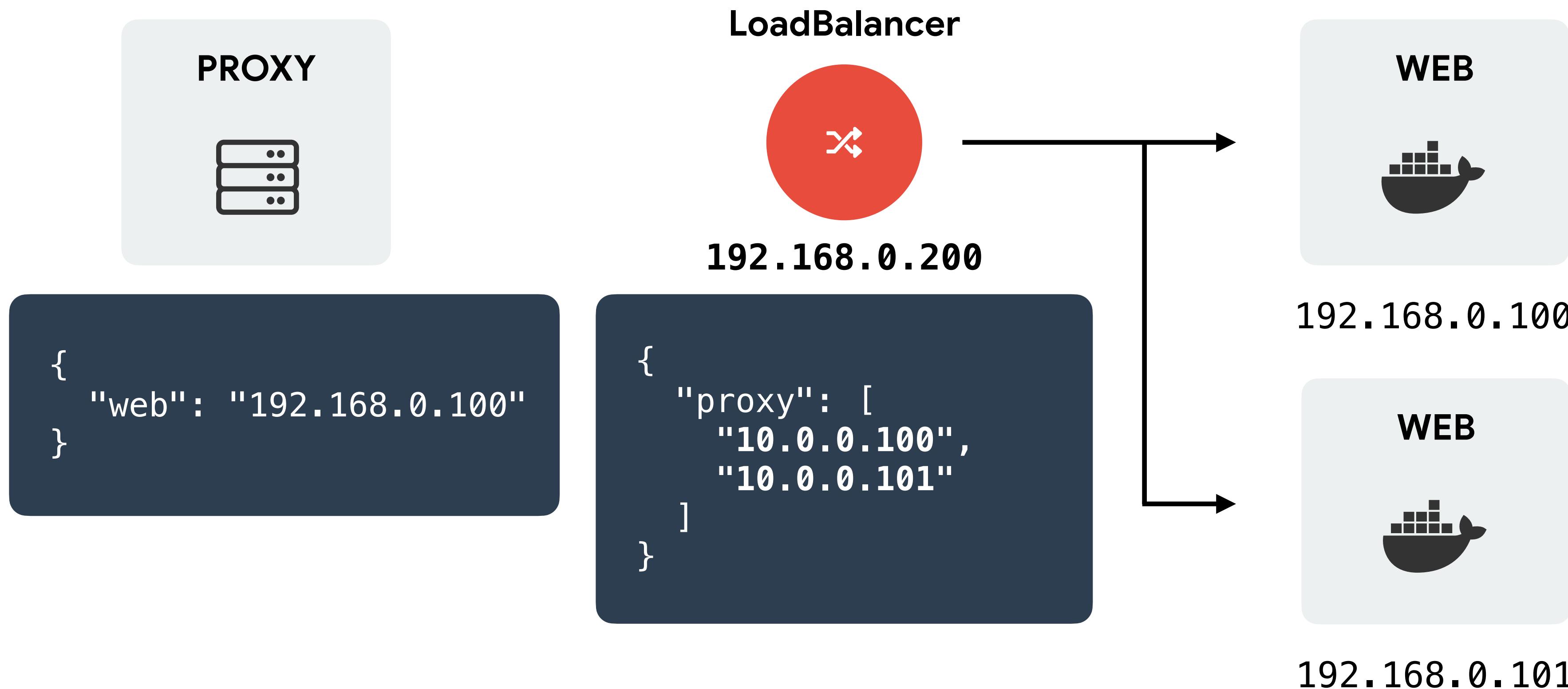
도커 그 이후



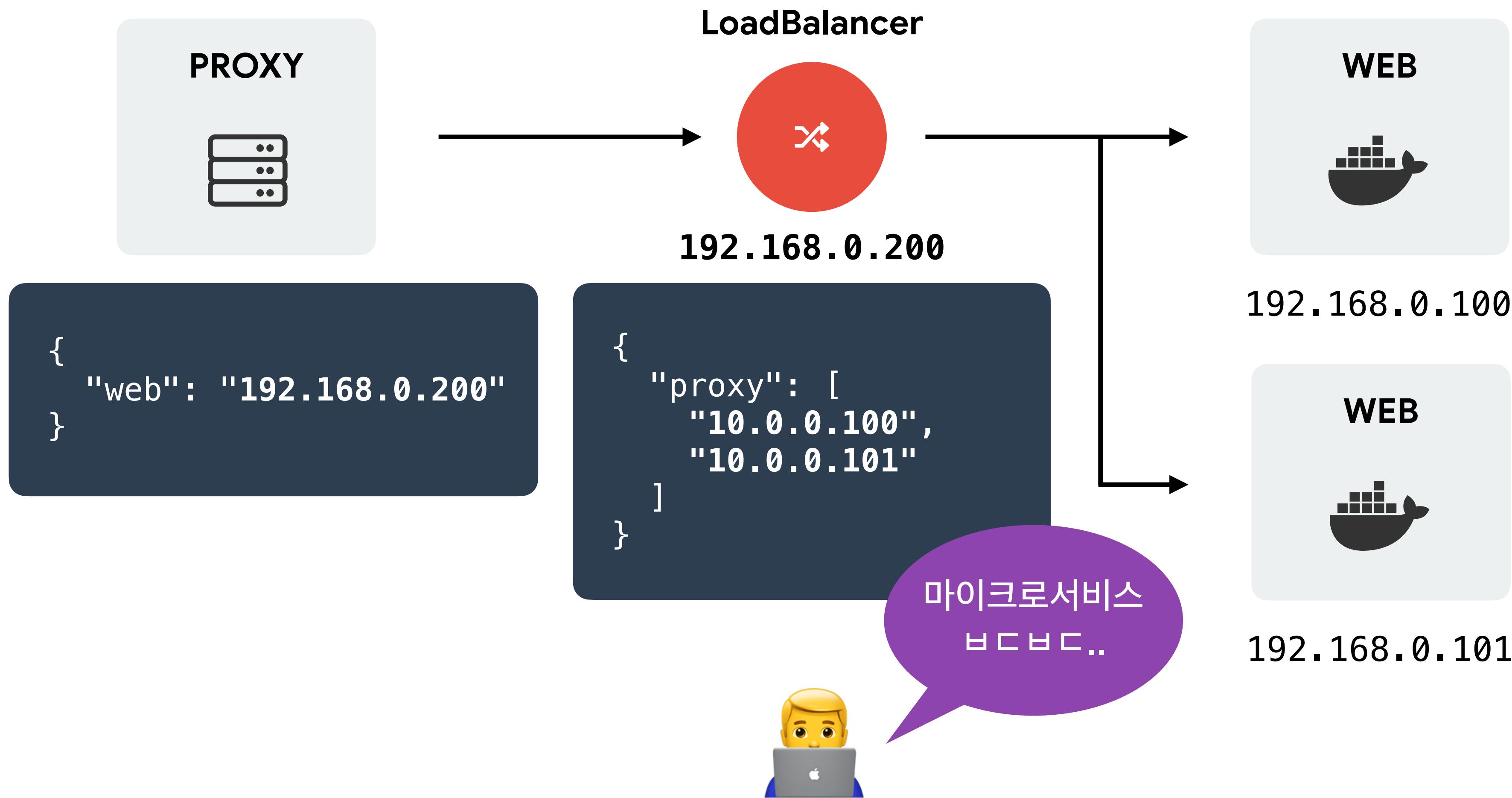
도커 그 이후



도커 그 이후



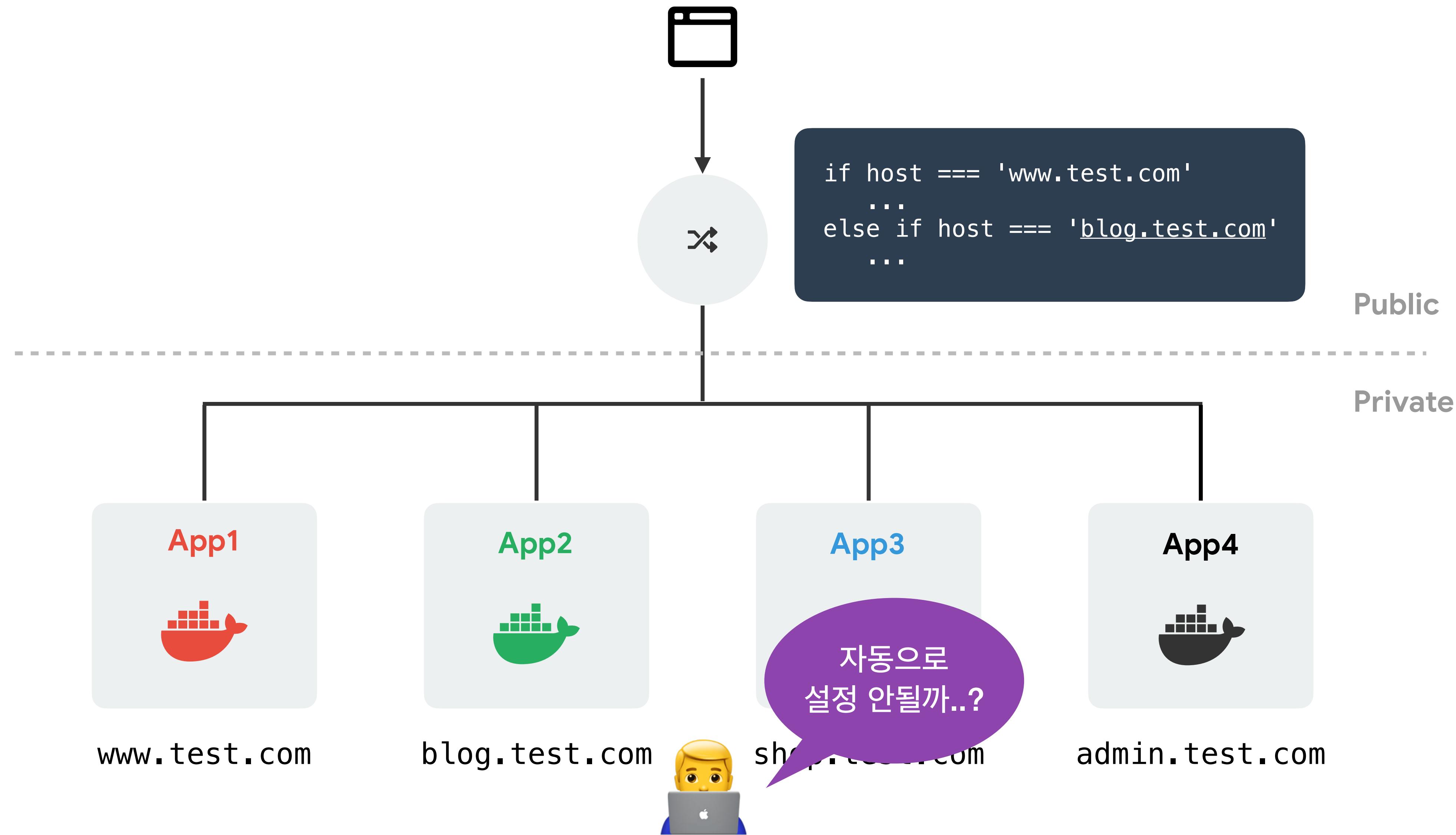
도커 그 이후



Gateway

3. 서비스 노출은 어떻게 할까?

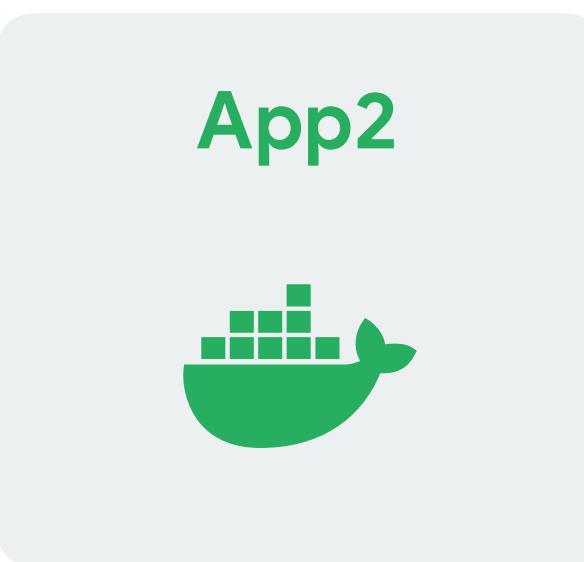
도커 그 이후



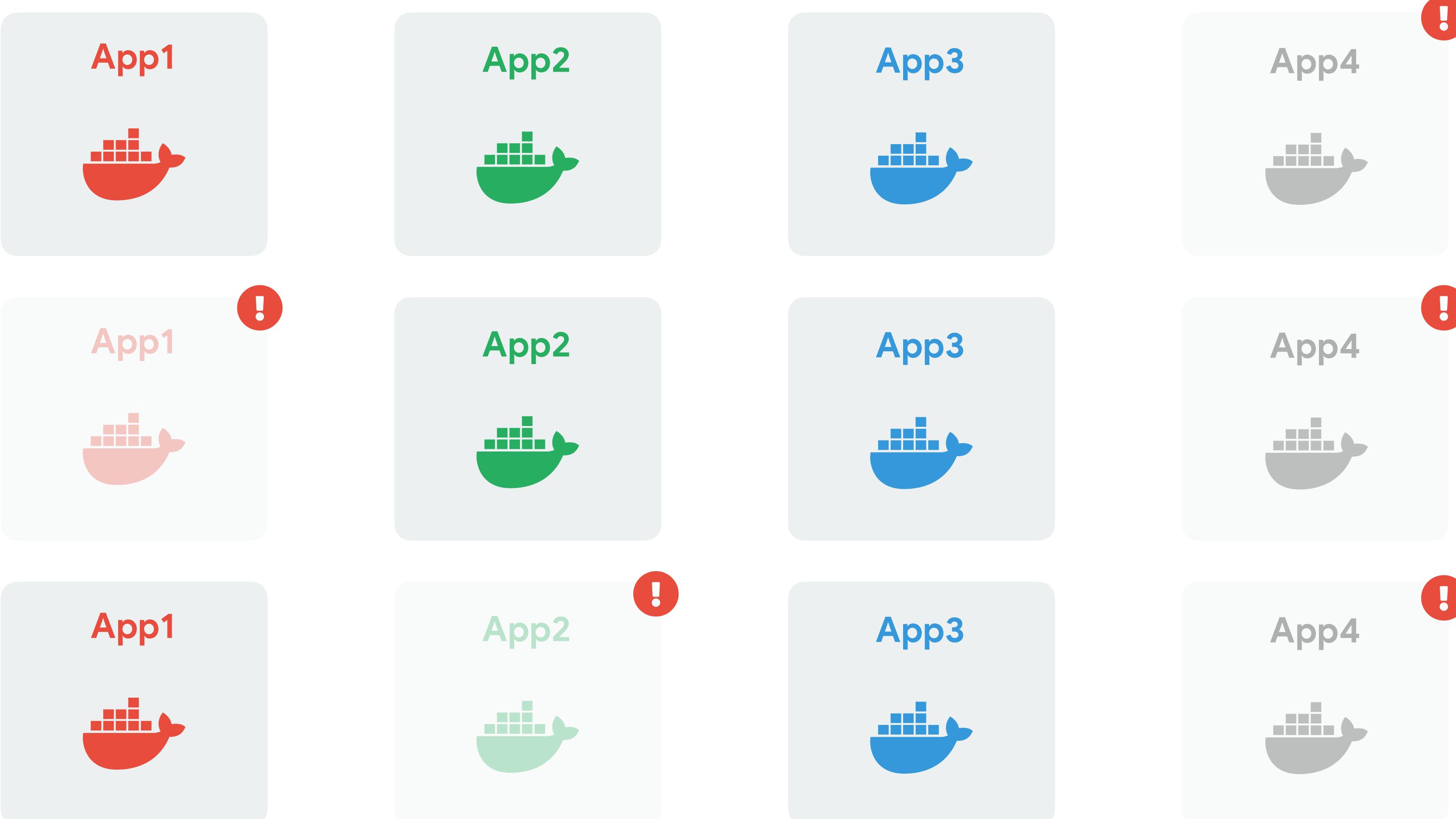
도커 그 이후

4. 서비스 이상, 부하 모니터링은 어떻게 할까?

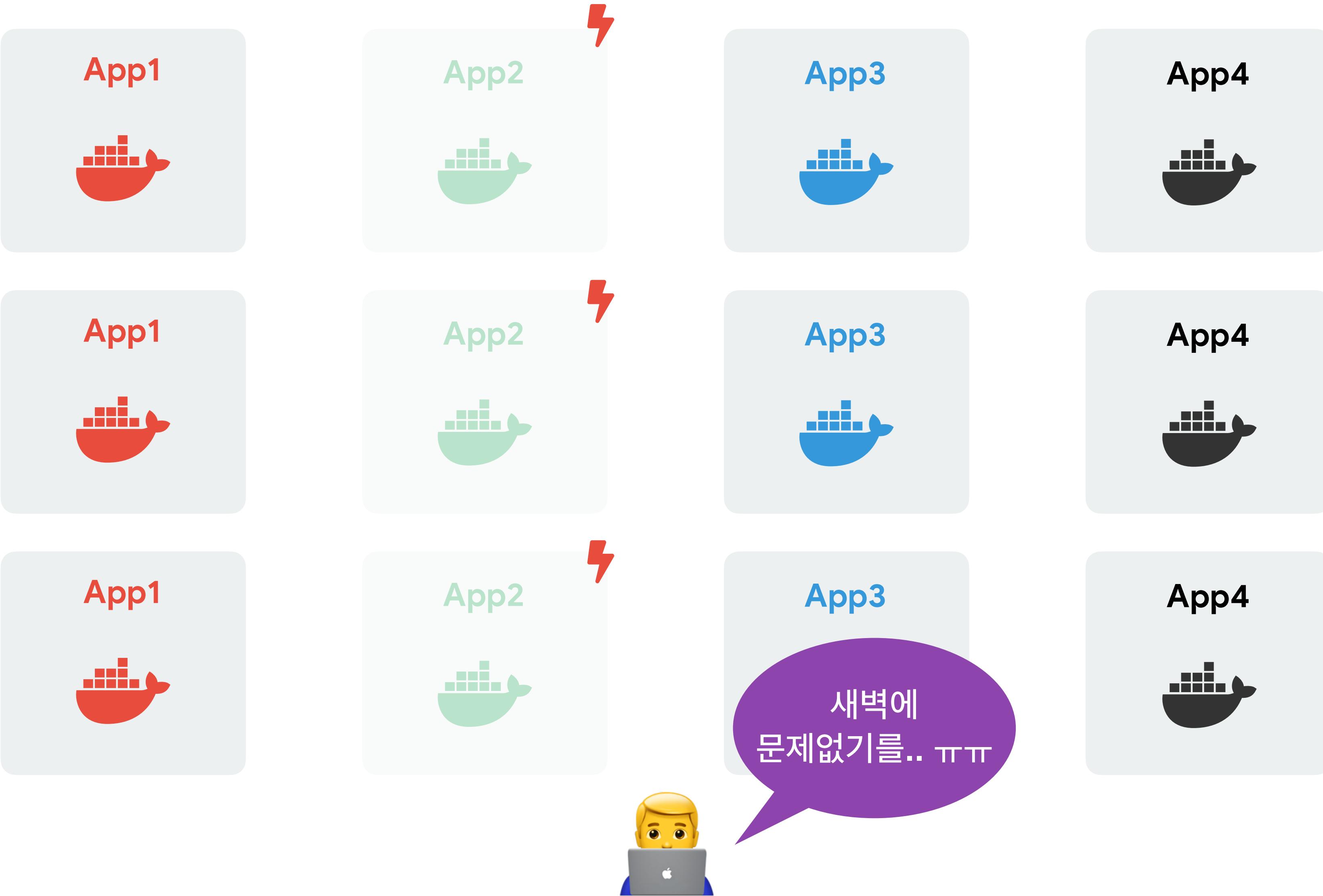
도커 그 이후



도커 그 이후



도커 그 이후



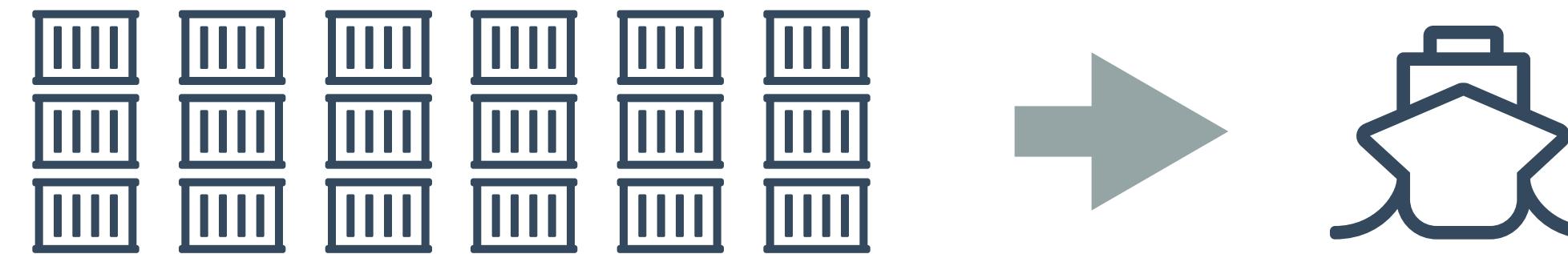
컨테이너 오케스트레이션



Container Orchestration

복잡한 컨테이너 환경을 효과적으로 관리하기 위한 도구

컨테이너 오케스트레이션



Container Orchestration

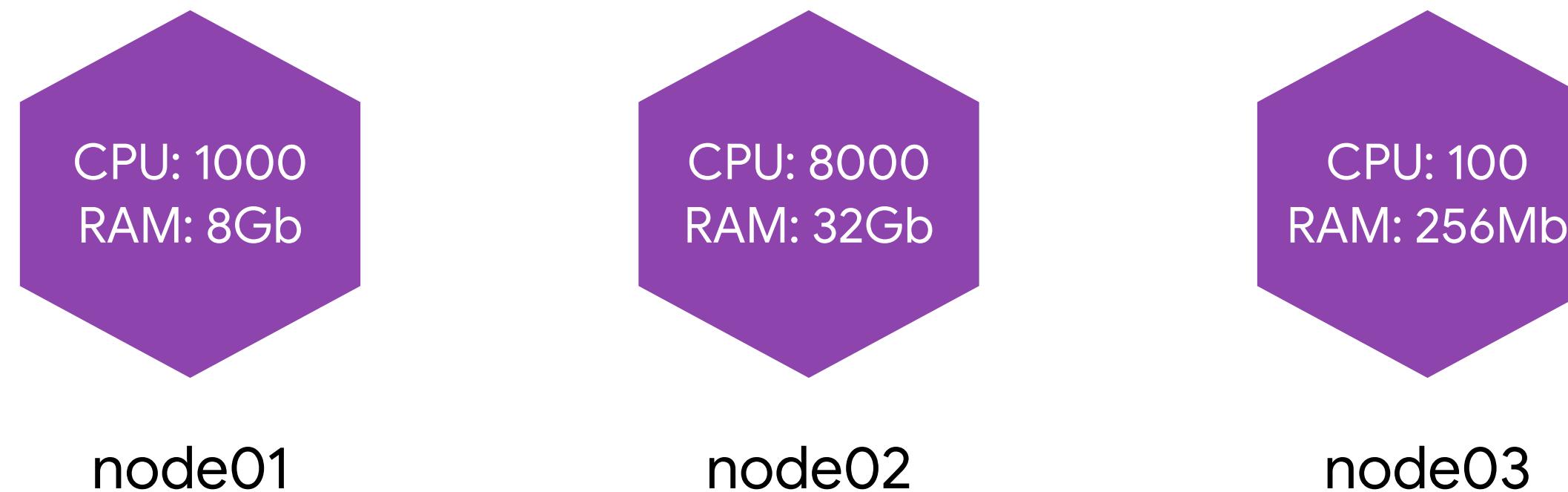
복잡한 컨테이너 환경을 효과적으로 관리하기 위한 도구



컨테이너 오케스트레이션



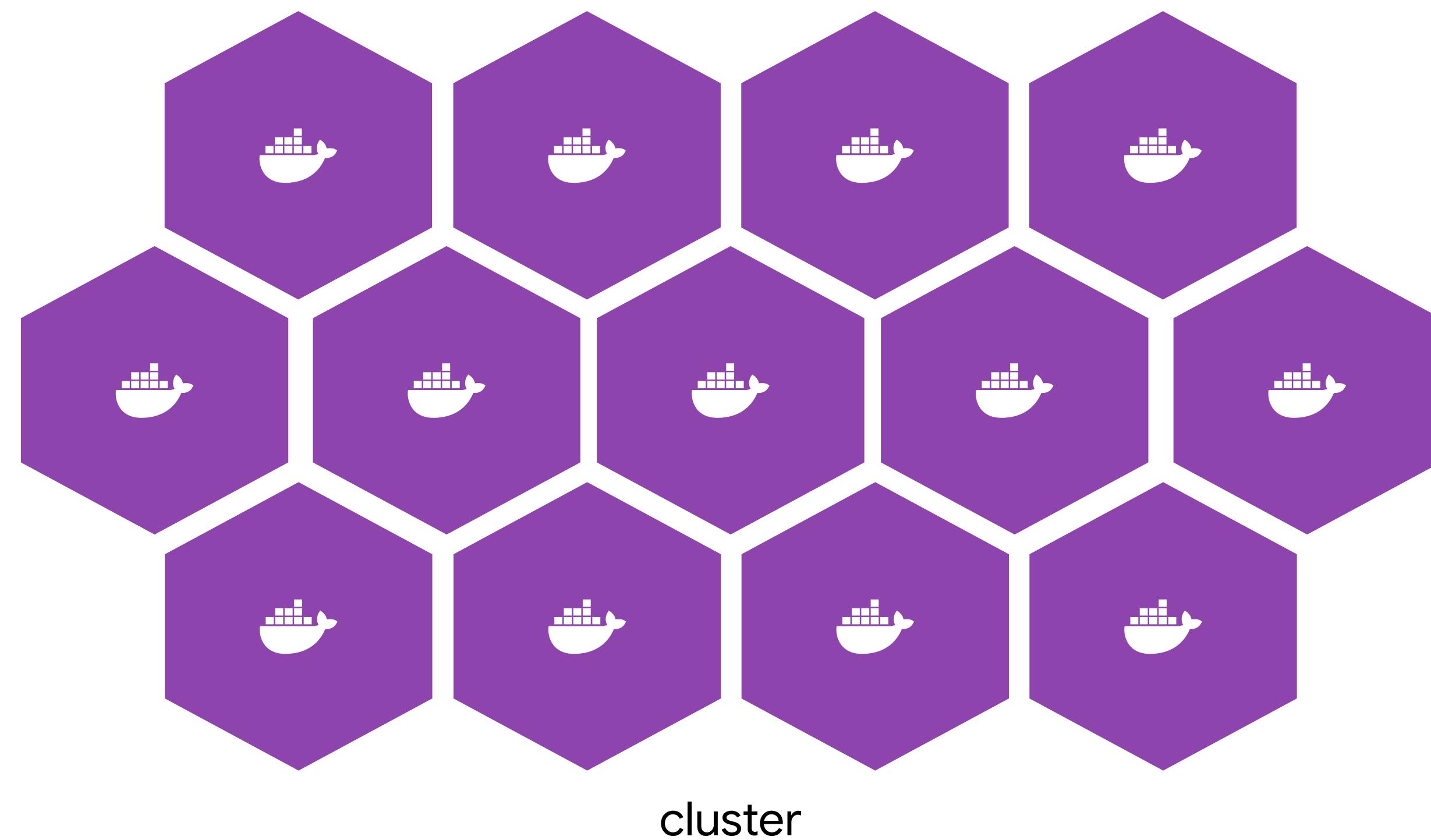
중앙제어 (master-node)



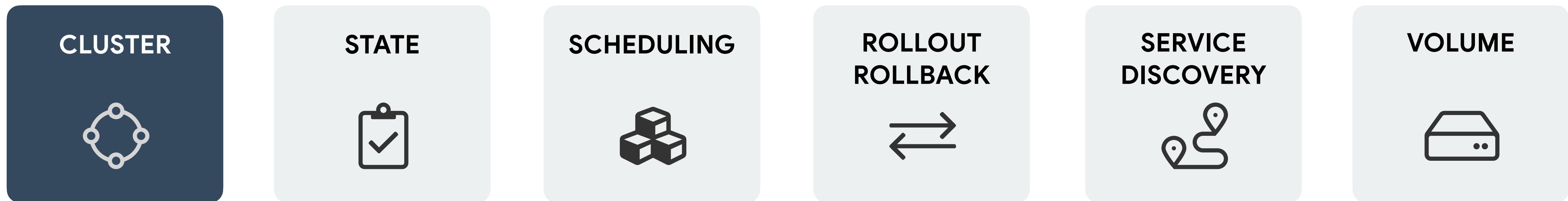
컨테이너 오케스트레이션



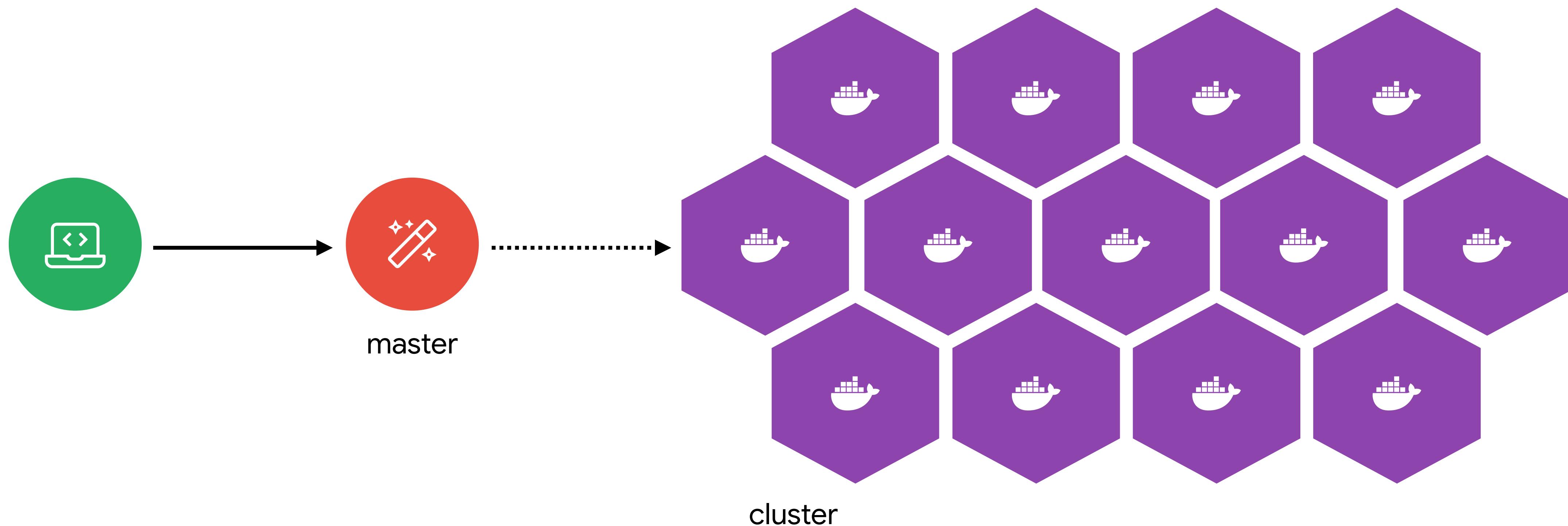
중앙제어 (master-node)



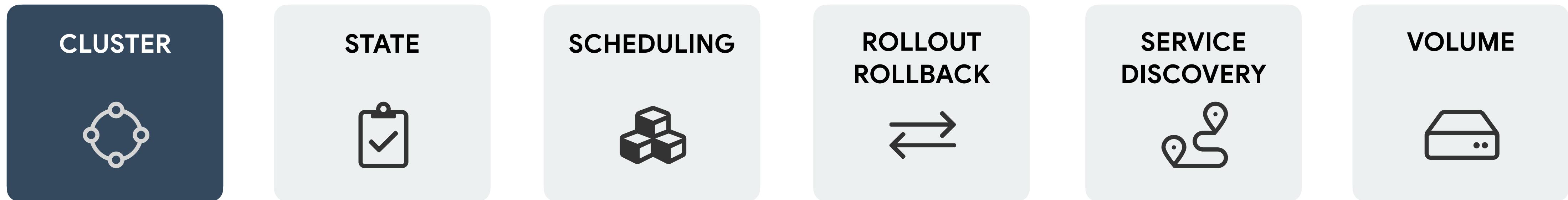
컨테이너 오케스트레이션



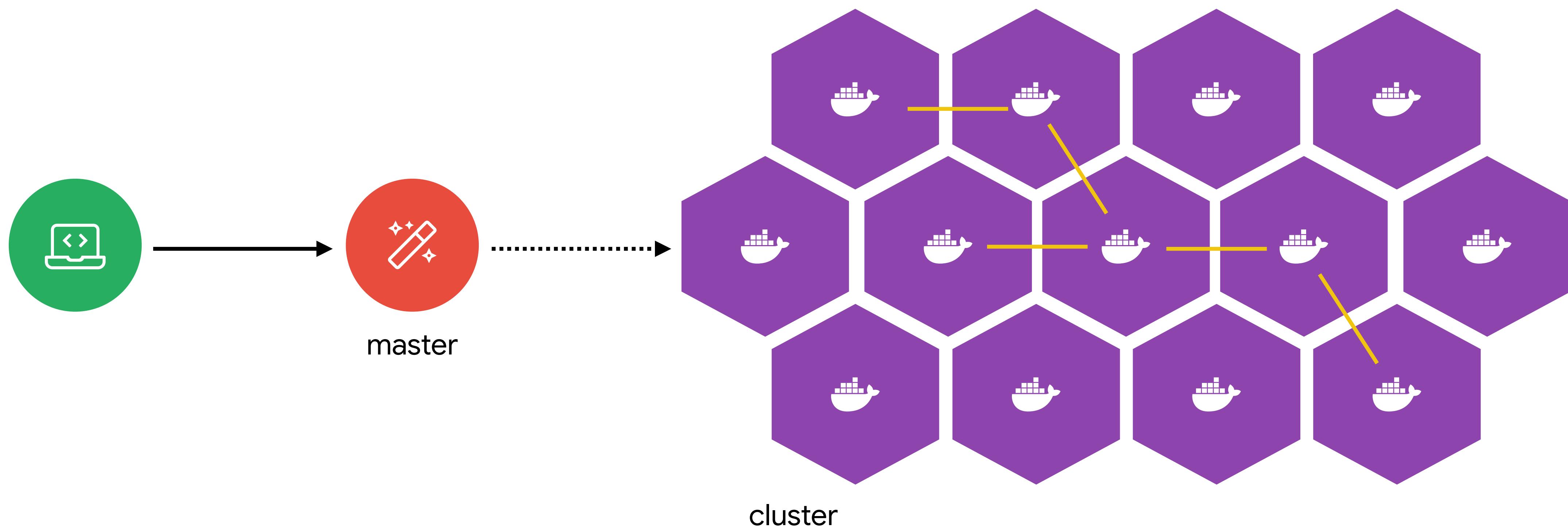
중앙제어 (master-node)



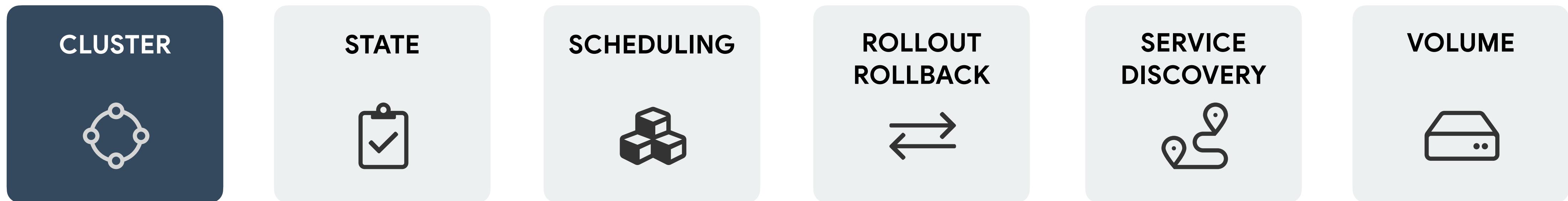
컨테이너 오케스트레이션



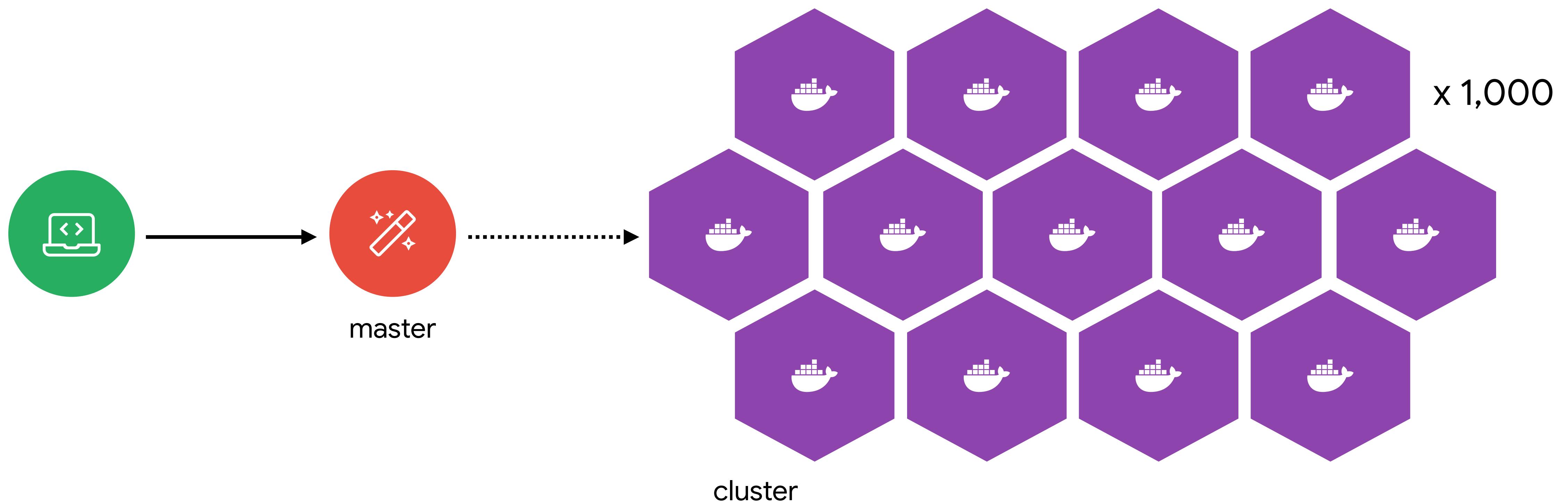
네트워킹



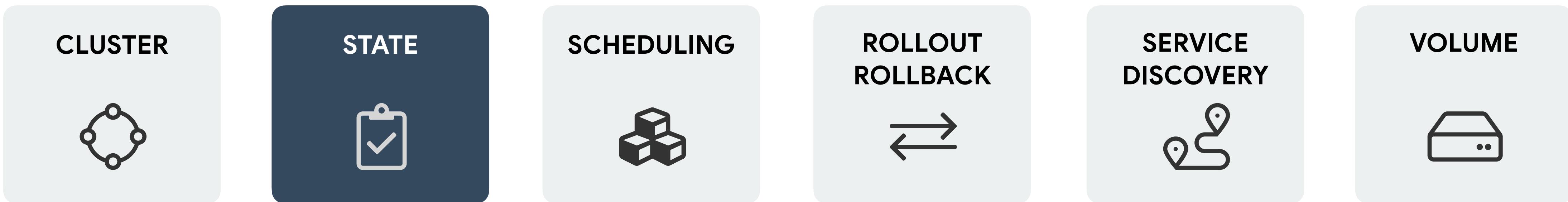
컨테이너 오케스트레이션



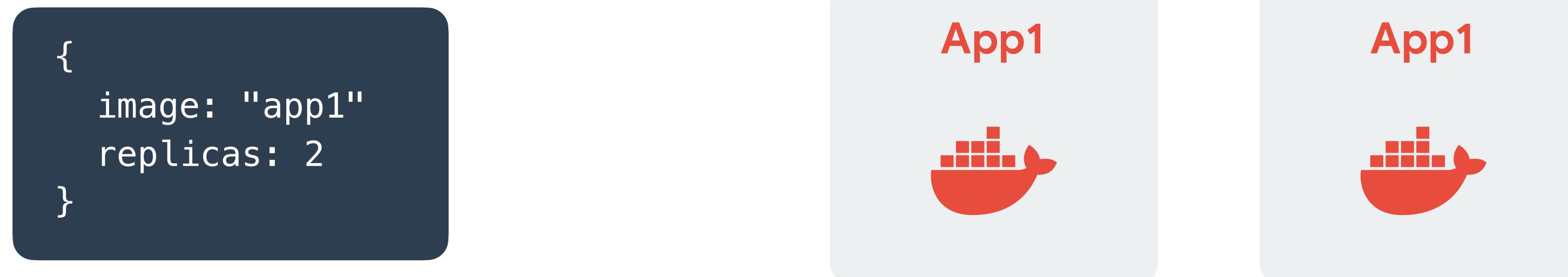
노드 스케일



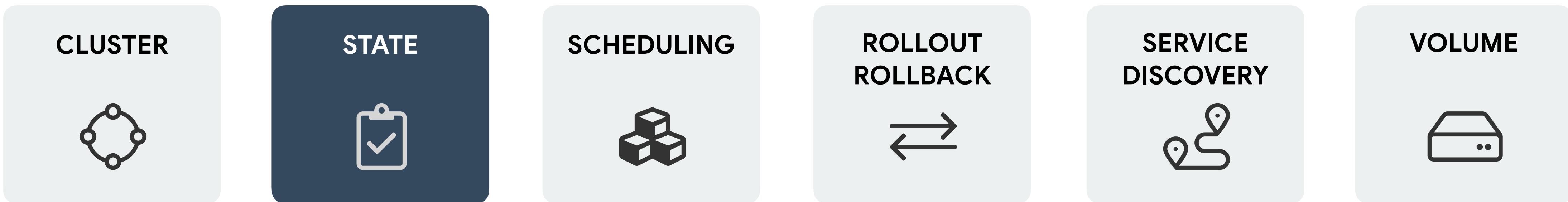
컨테이너 오케스트레이션



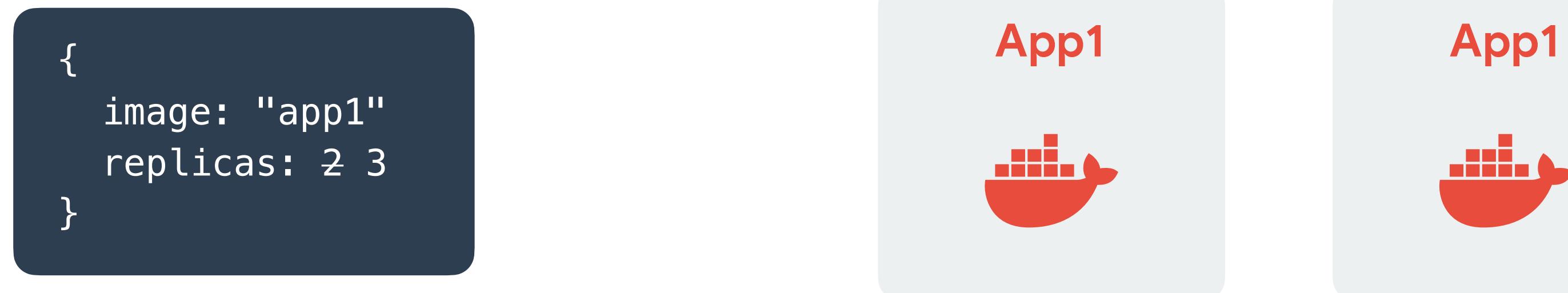
상태 관리



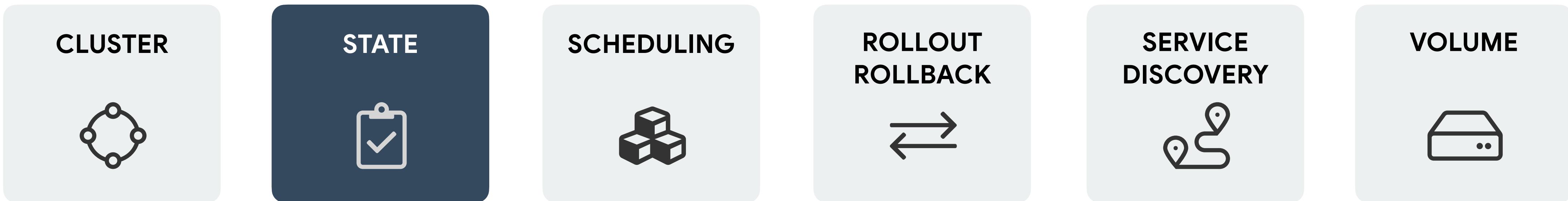
컨테이너 오케스트레이션



상태 관리

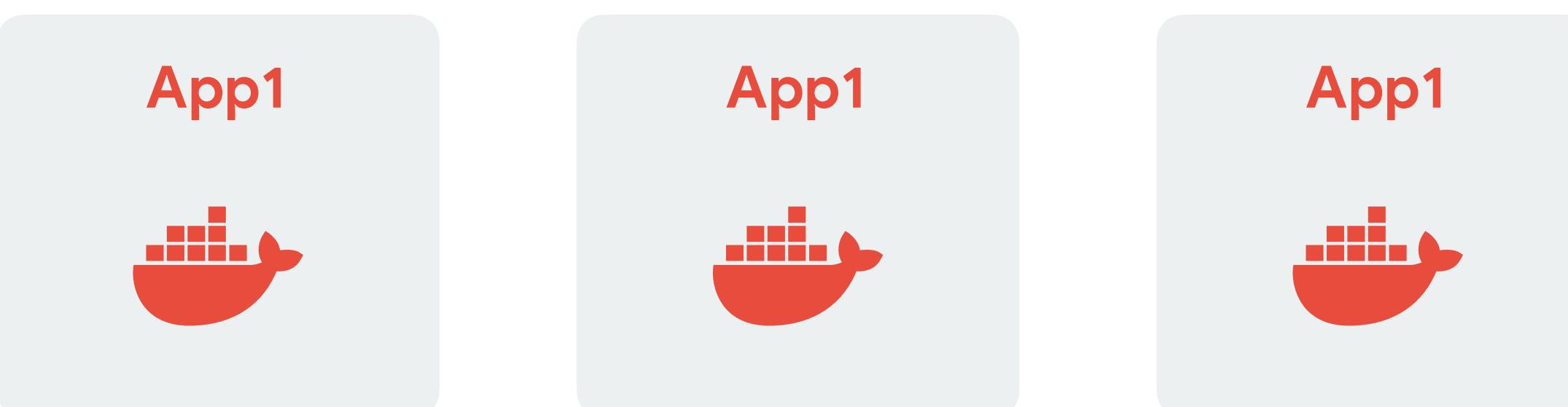


컨테이너 오케스트레이션

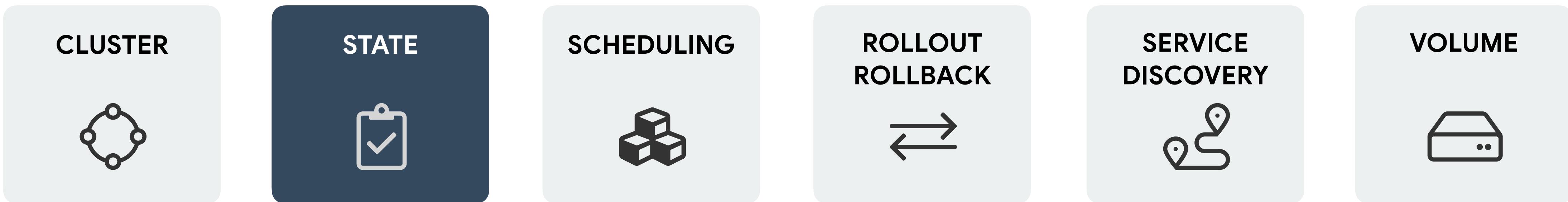


상태 관리

```
{  
  image: "app1"  
  replicas: 3  
}
```

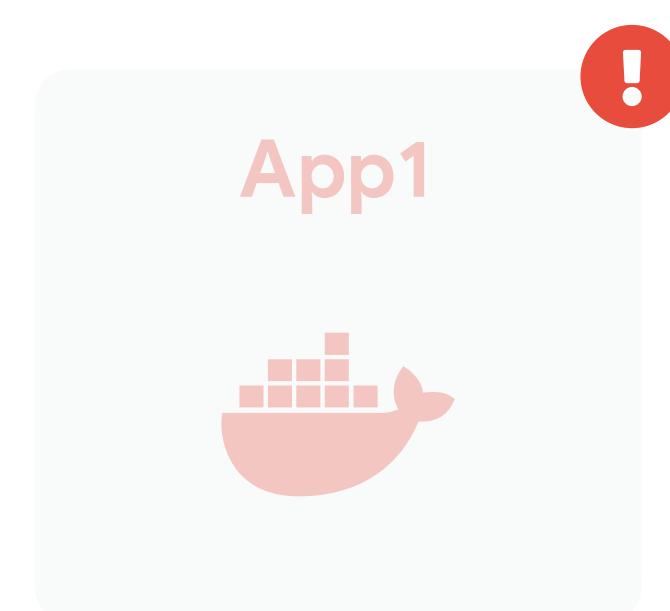
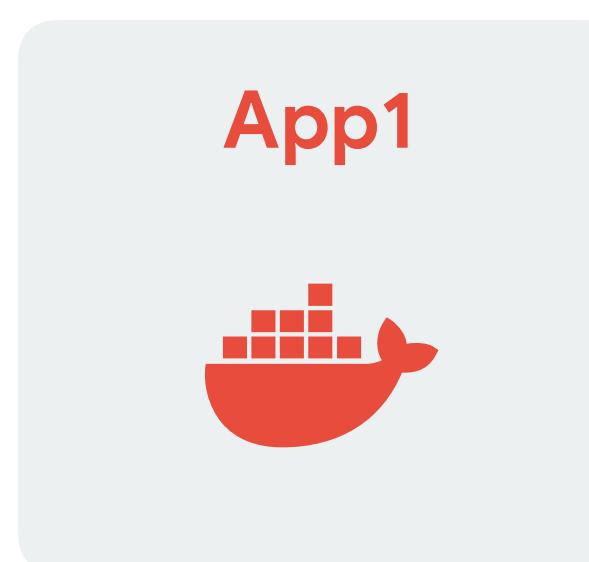


컨테이너 오케스트레이션

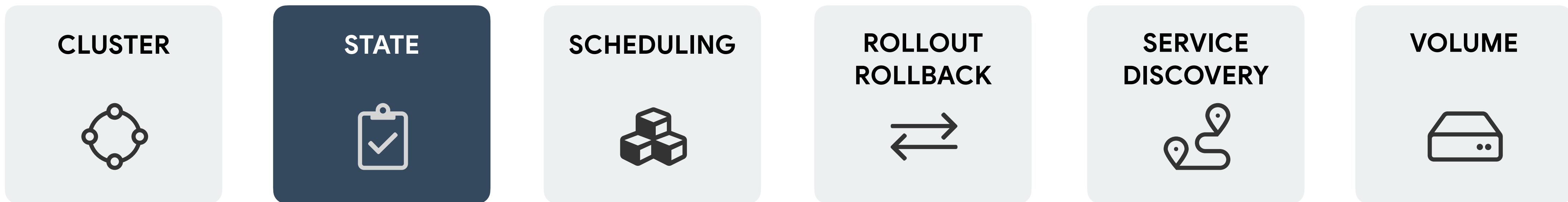


상태 관리

```
{  
  image: "app1"  
  replicas: 3  
}
```

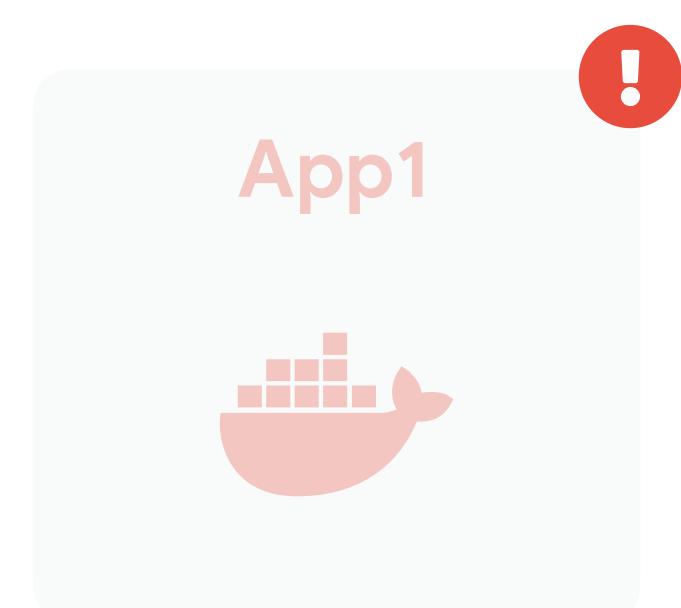


컨테이너 오케스트레이션

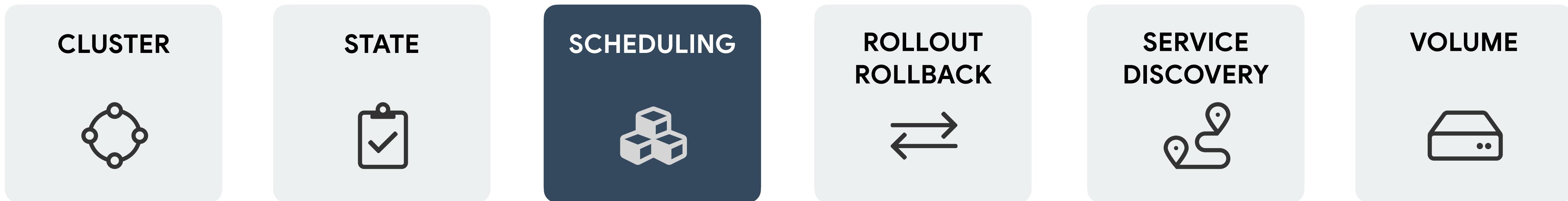


상태 관리

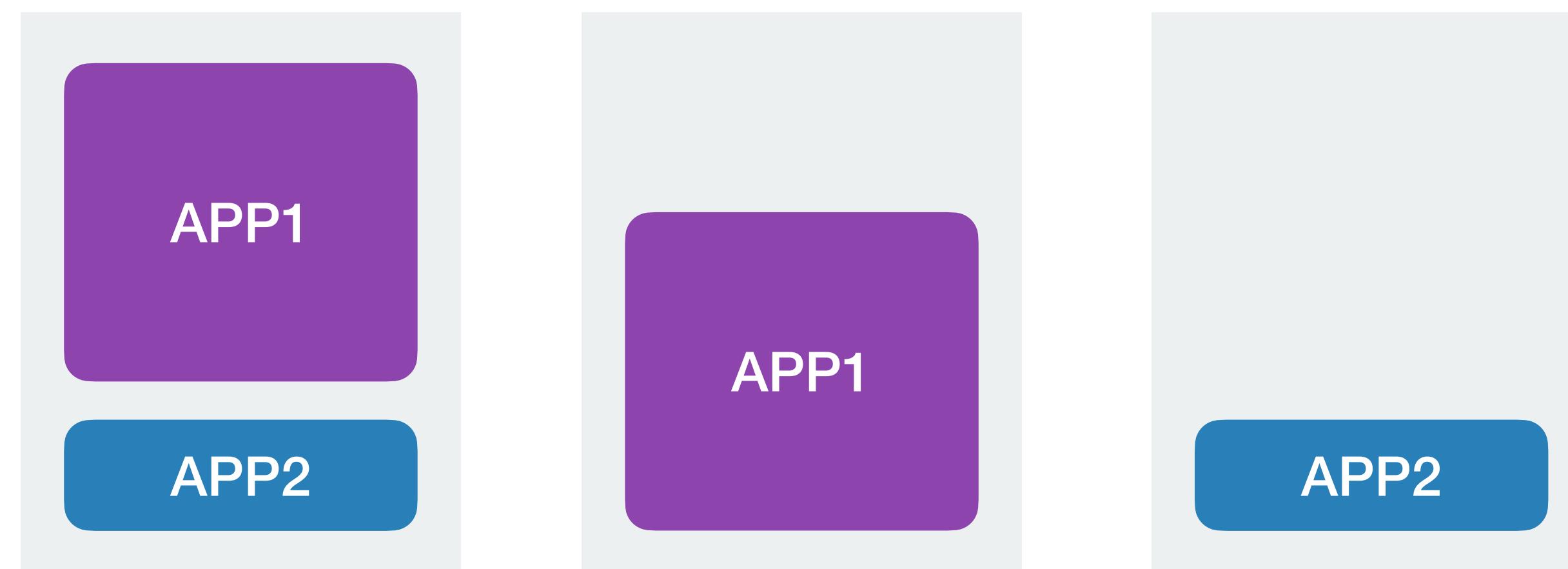
```
{  
  image: "app1"  
  replicas: 3  
}
```



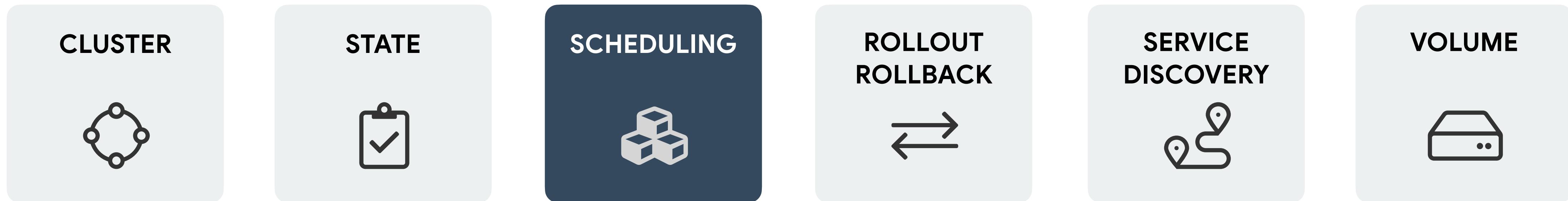
컨테이너 오케스트레이션



배포 관리



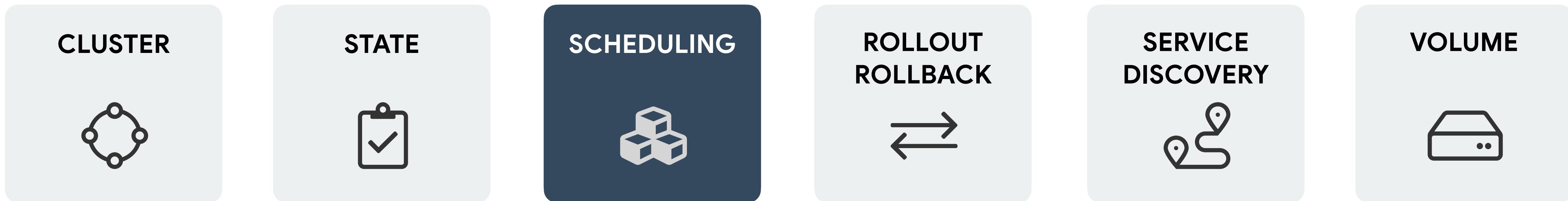
컨테이너 오케스트레이션



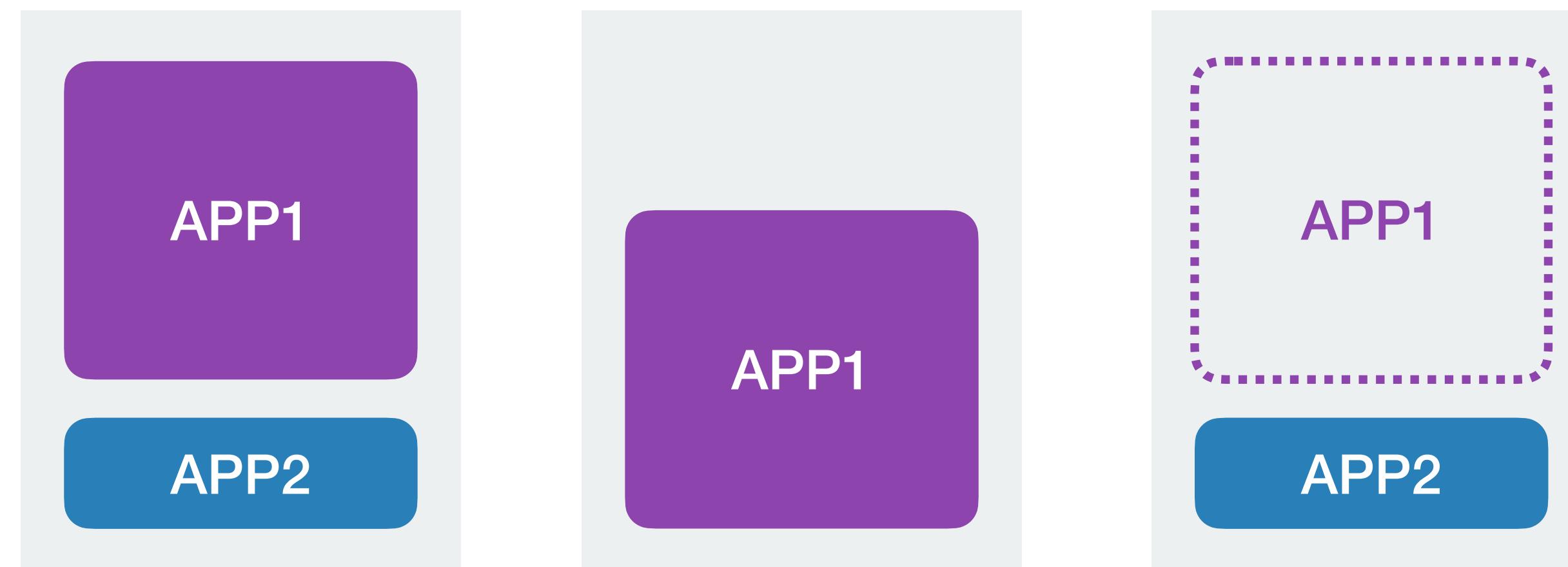
배포 관리



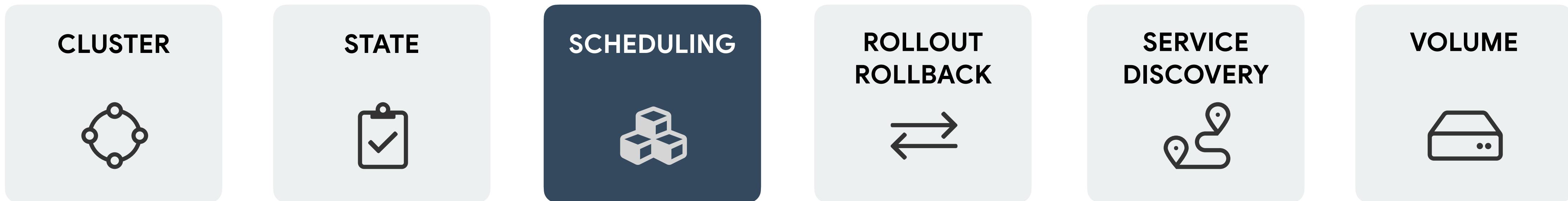
컨테이너 오케스트레이션



배포 관리



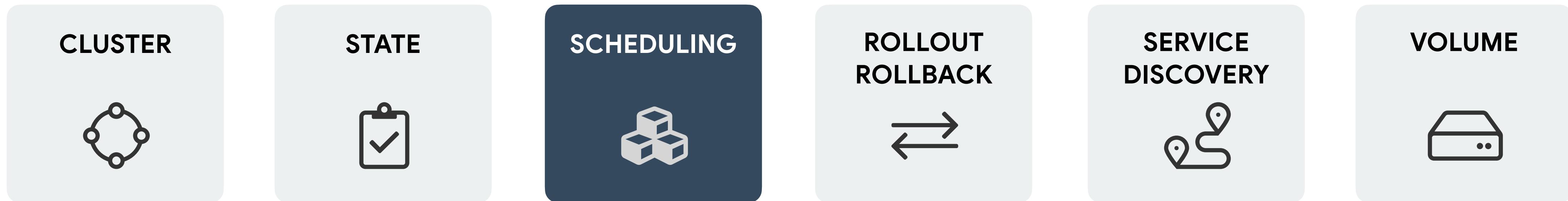
컨테이너 오케스트레이션



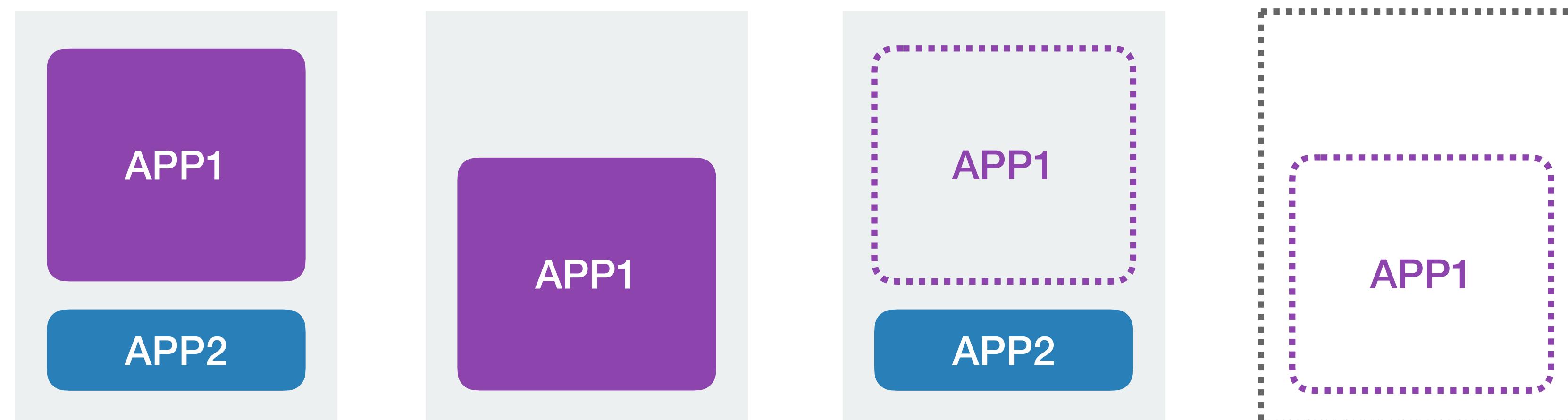
배포 관리



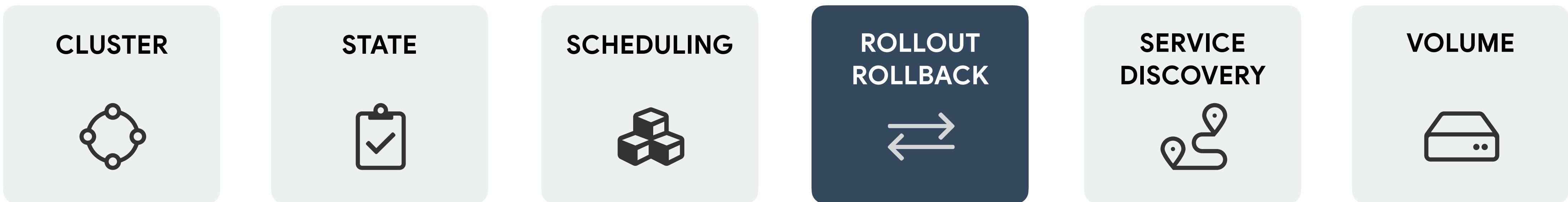
컨테이너 오케스트레이션



배포 관리



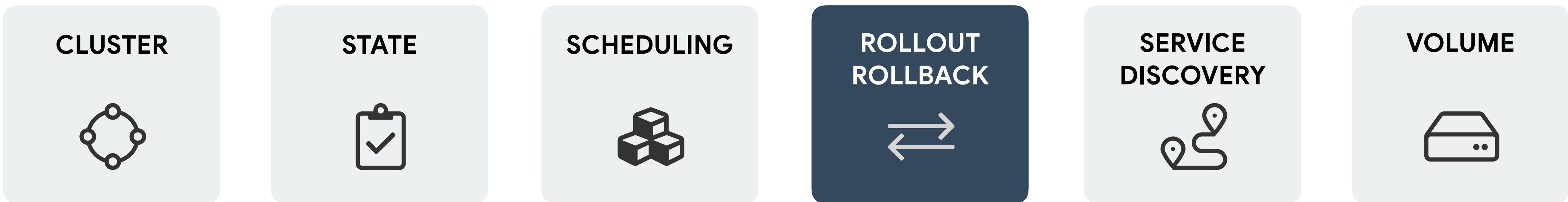
컨테이너 오케스트레이션



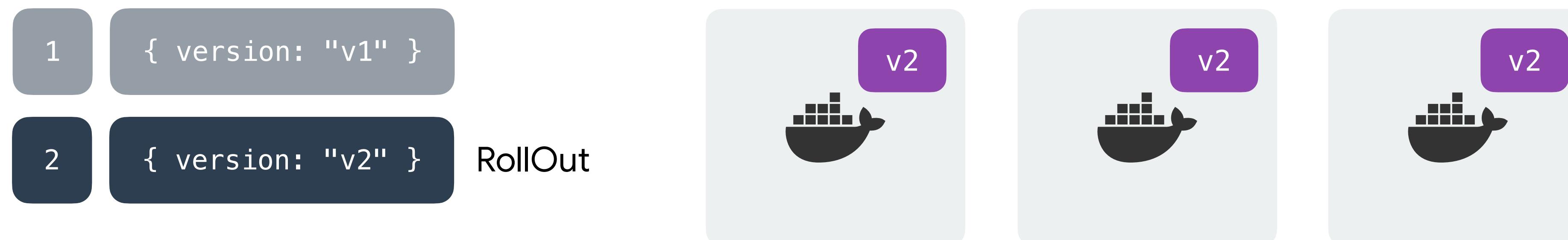
배포 버전관리



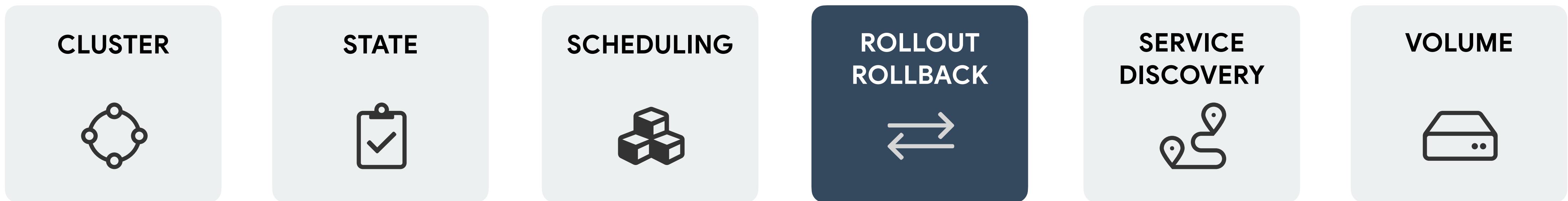
컨테이너 오케스트레이션



배포 버전관리



컨테이너 오케스트레이션



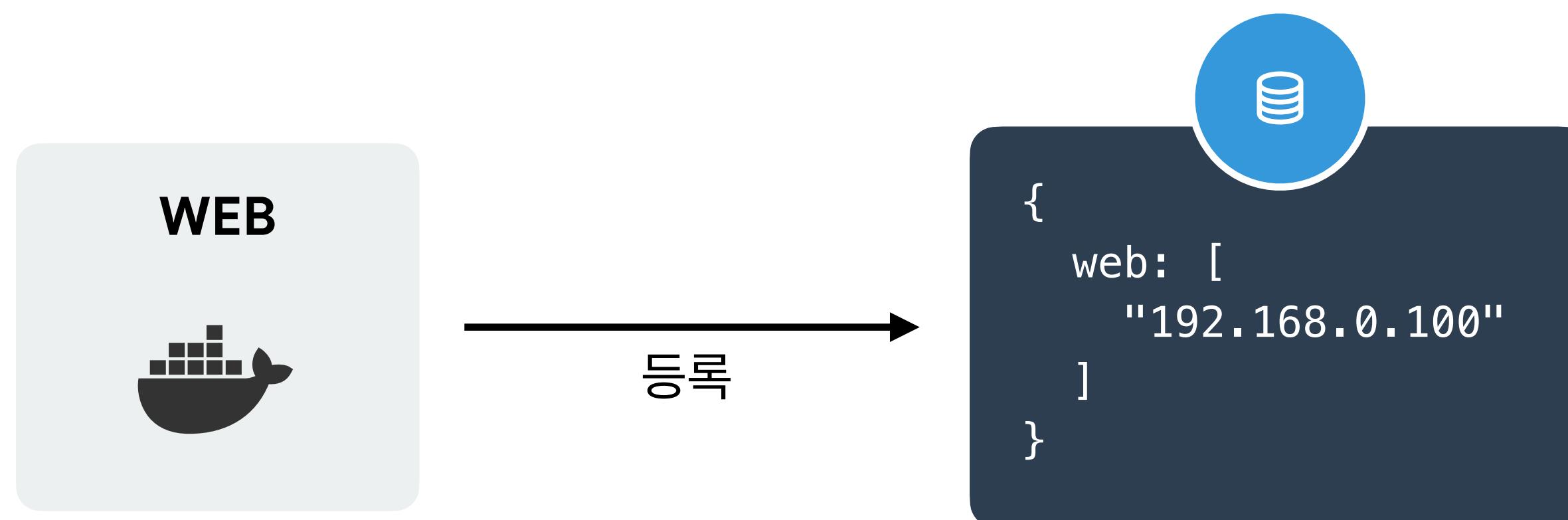
배포 버전관리



컨테이너 오케스트레이션

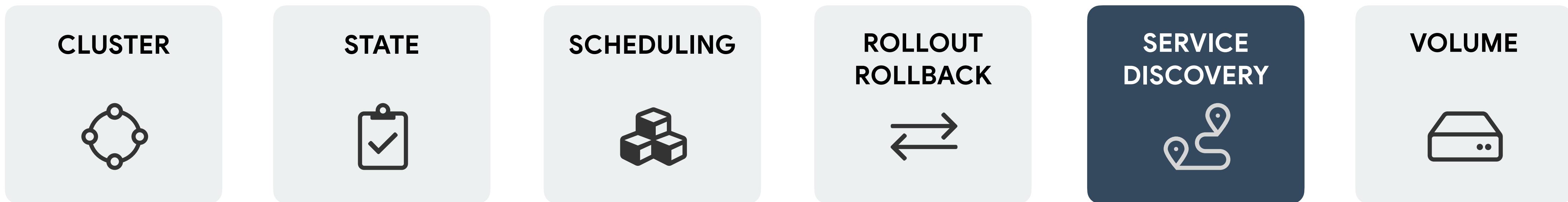


서비스 등록 및 조회

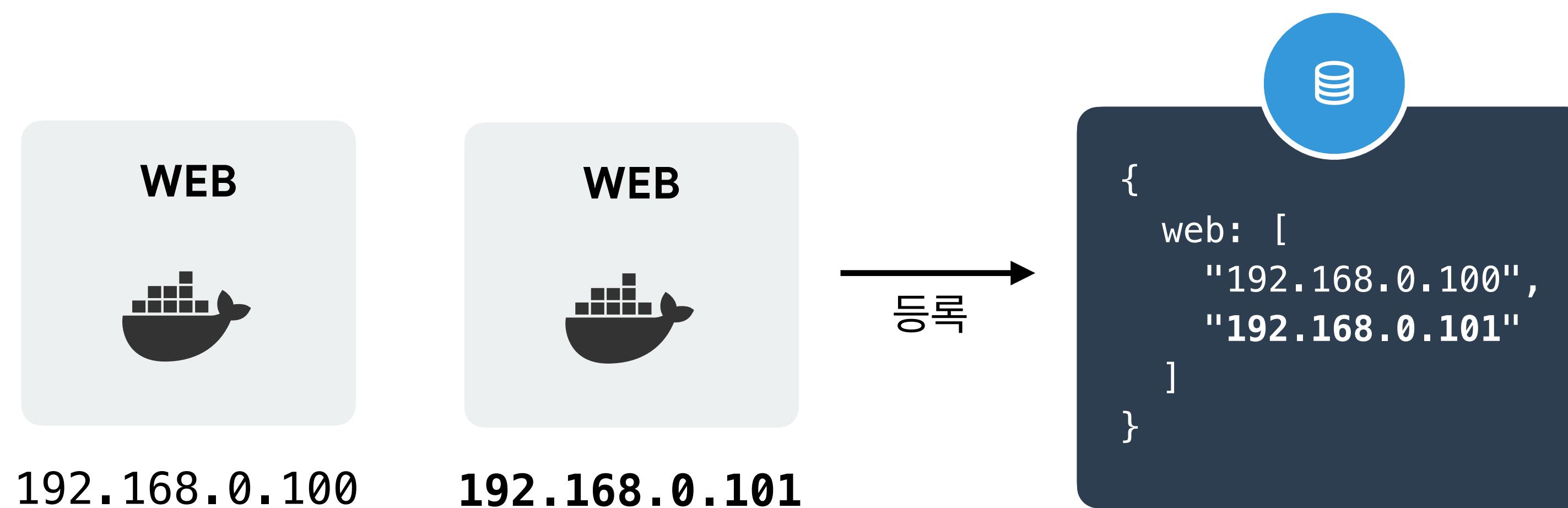


192.168.0.100

컨테이너 오케스트레이션



서비스 등록 및 조회



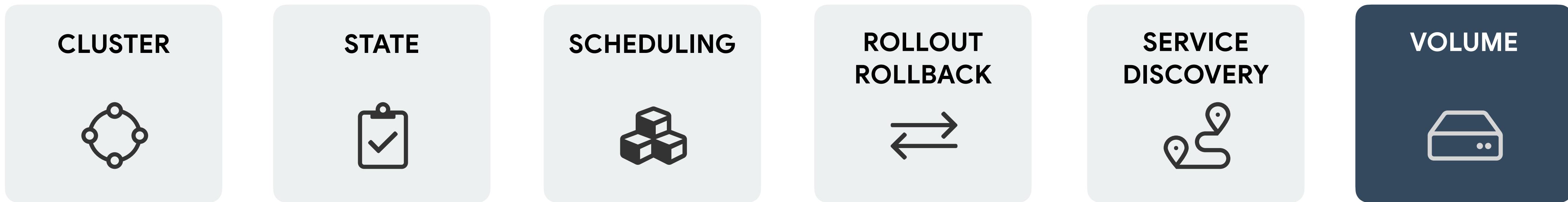
컨테이너 오케스트레이션



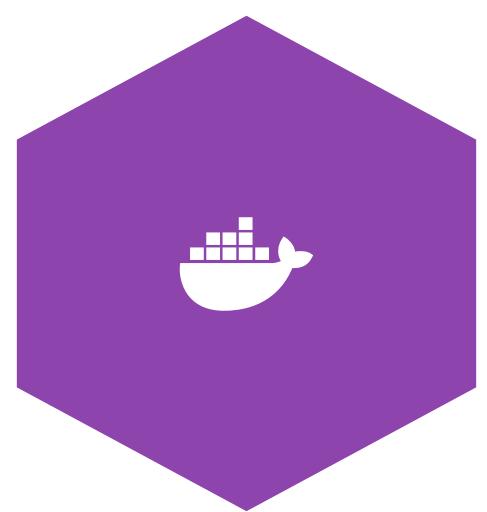
서비스 등록 및 조회



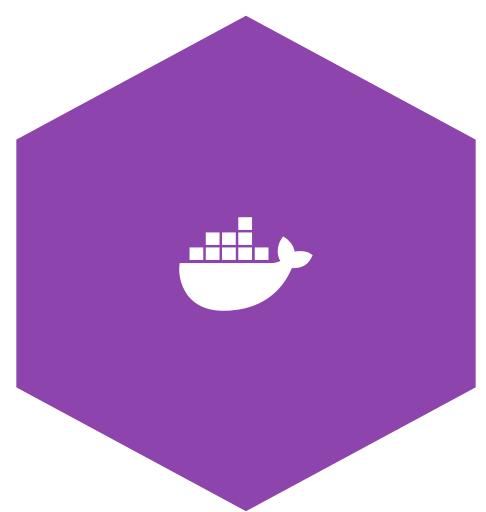
컨테이너 오케스트레이션



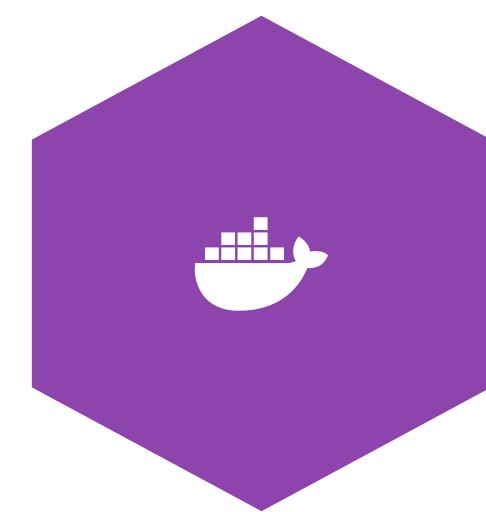
볼륨 스토리지



node01

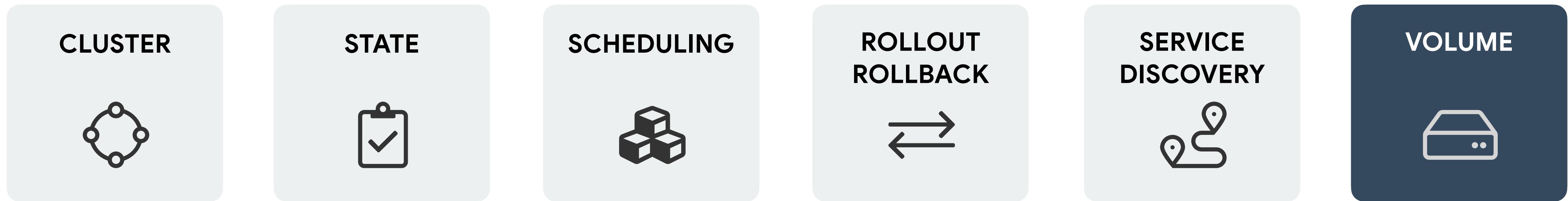


node02

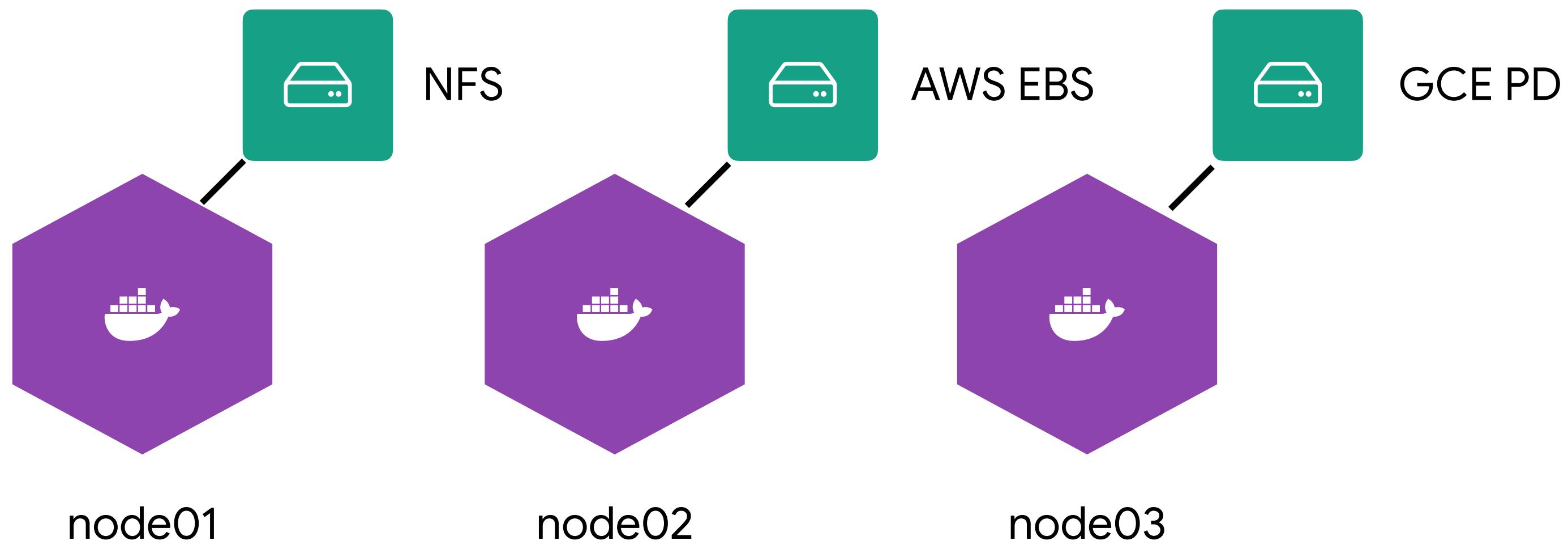


node03

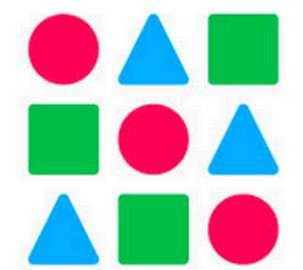
컨테이너 오케스트레이션



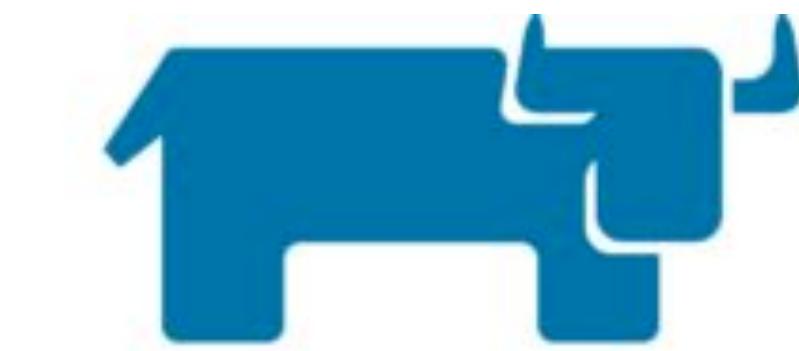
볼륨 스토리지



컨테이너 오케스트레이션



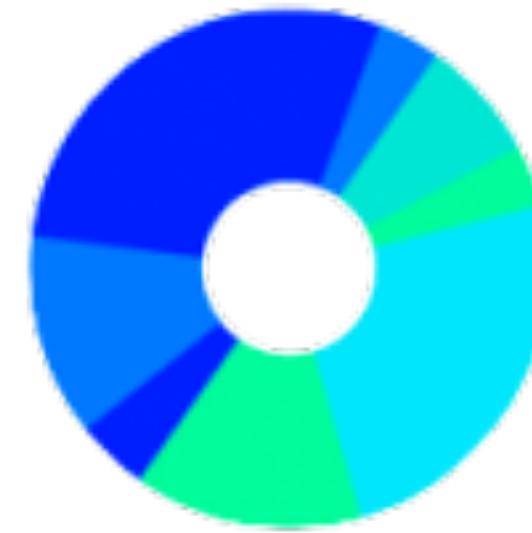
DEIS



RANCHER



MESOS MARATHON



HashiCorp
Nomad

컨테이너 오케스트레이션



도커처럼 사용하기 쉬운
컨테이너 오케스트레이션!

컨테이너 관리도구 춘추전국시대



왜 쿠버네티스인가?



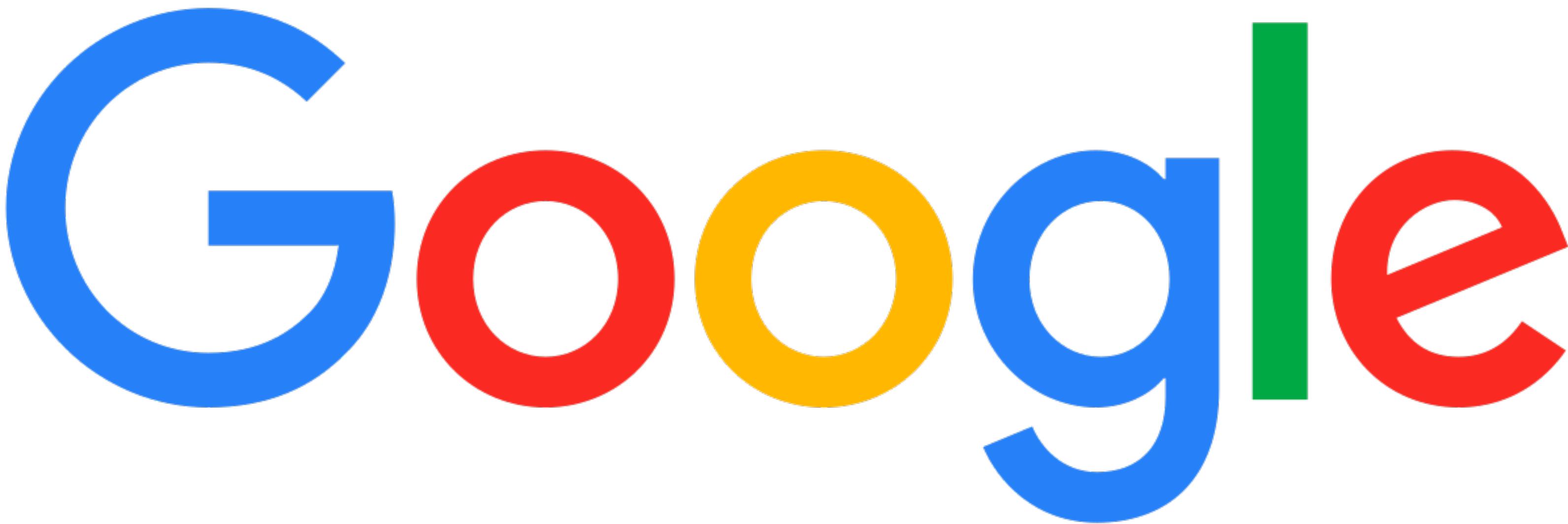
쿠버네티스 소개



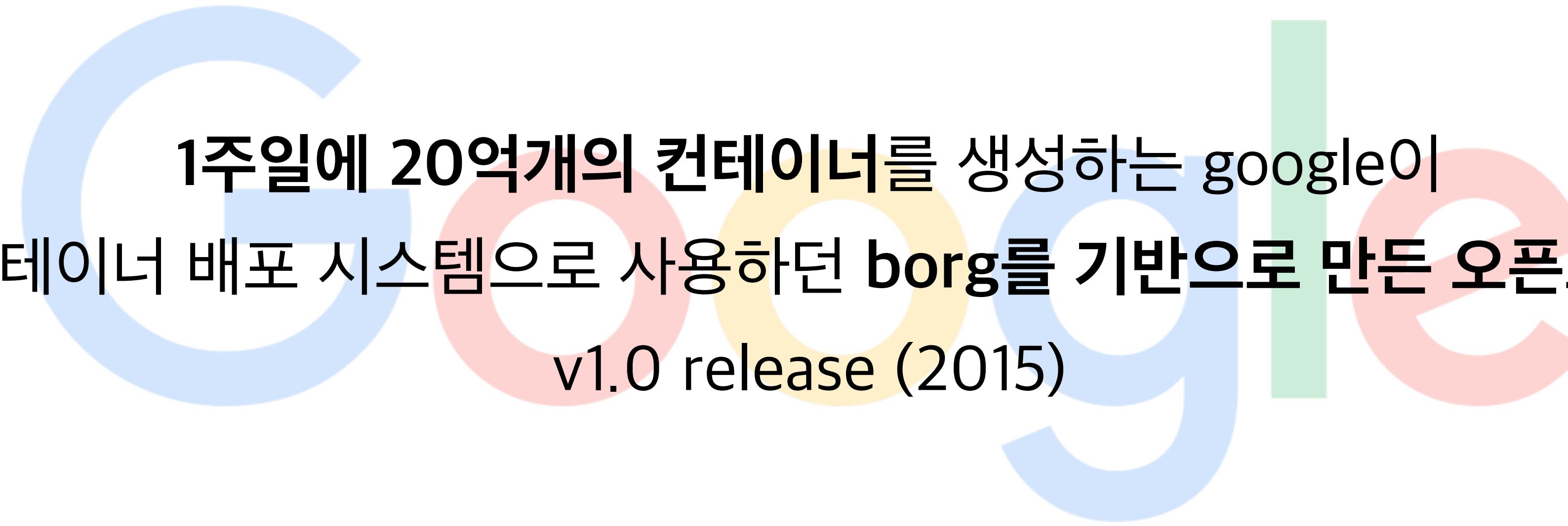
kubernetes

컨테이너를 쉽고 빠르게 배포/확장하고 관리를 자동화해주는 오픈소스 플랫폼

쿠버네티스 소개



쿠버네티스 소개



1주일에 20억개의 컨테이너를 생성하는 google의
컨테이너 배포 시스템으로 사용하던 borg를 기반으로 만든 오픈소스
v1.0 release (2015)

쿠버네티스 소개

1주일에 20억개의 컨테이너를 생성하는 google이
컨테이너 배포 시스템으로 사용하던 borg를 기반으로 만든 오픈소스
v1.0 release (2015)



쿠버네티스 소개

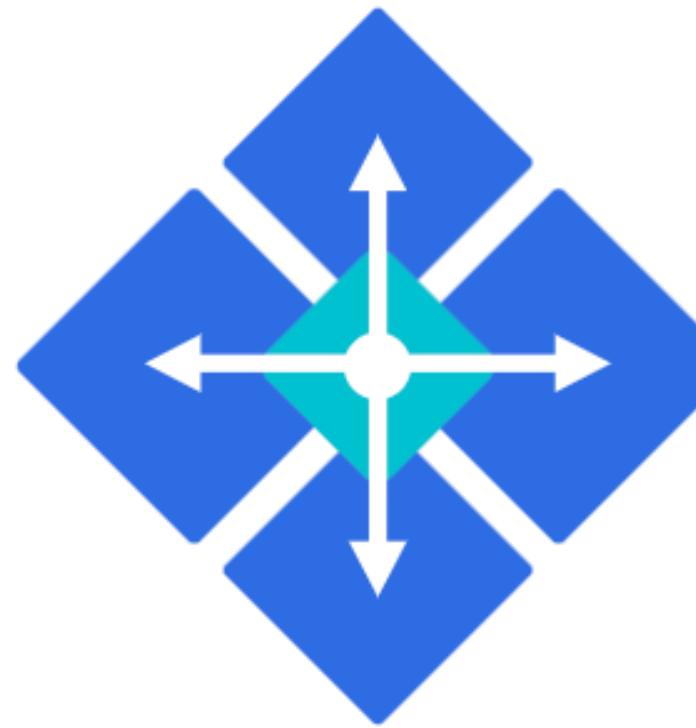
Production-Grade Container Orchestration

Automated container deployment, scaling, and management

[Learn Kubernetes Basics](#)

운영에서 사용가능한 컨테이너 오케스트레이션

쿠버네티스 소개



Planet Scale

Designed on the same principles that allows Google to run billions of containers a week, Kubernetes can scale without increasing your ops team.

행성 스케일 구글보다 저게 사용하면 OK

쿠버네티스 소개

Never Outgrow

Whether testing locally or running a global enterprise, Kubernetes flexibility grows with you to deliver your applications consistently and easily no matter how complex your need is.



다양한 요구사항을 만족시킬 수 있는 유연함

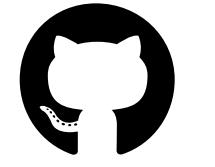
쿠버네티스 소개

Run Anywhere

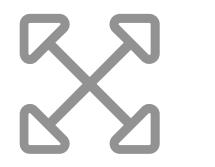
Kubernetes is open source giving you the freedom to take advantage of on-premises, hybrid, or public cloud infrastructure, letting you effortlessly move workloads to where it matters to you.

어디서나 동작

왜 쿠버네티스인가

 오픈소스

 엄청난 인기

 무한한 확장성

 사실상의 표준 (de facto)

오픈소스

 [kubernetes / kubernetes](#)

Watch 3.3k Unstar 69.7k Fork 25.2k

< Code Issues 2k Pull requests 877 Actions Projects 8 Security Insights

master ▾ 42 branches 698 tags Go to file Add file ▾ Code ▾

 **k8s-ci-robot** Merge pull request #94112 from damemi/... ... ✓ 1f708f6 6 hours ago 93,927 commits

| | | |
|---|--|-------------|
|  .github | Add a link to liggitt's deflake docs in Flake template (#942...) | 8 days ago |
|  CHANGELOG | Merge pull request #93975 from avr1254/master | 2 days ago |
|  LICENSES | Fix broken dependency vbom.ml/util | 2 days ago |
|  api | Merge pull request #94010 from ghostsquad/docs/hpa-v2... | yesterday |
|  build | Stop ignoring unit test flakes, require multiple successful ... | yesterday |
|  cluster | Merge pull request #93975 from avr1254/master | 2 days ago |
|  cmd | Merge pull request #94535 from liggitt/kubeadm-unit-fix | 8 hours ago |
|  docs | Add sigs for root folders | 23 days ago |

About

Production-Grade Container
Scheduling and Management

 [kubernetes.io](#)

kubernetes go cncf
containers

 Readme

 Apache-2.0 License

Releases 698

 v1.16.15 Latest
3 days ago

Google, Red Hat, Huawei, VMware, Microsoft, IBM, Intel, ...



The Kubernetes community -- users, contributors, and the culture we've built together -- is one of the biggest reasons for the meteoric rise of this open source project. Our culture and values continue to grow and change as the project itself grows and changes. We all work together toward constant improvement of the project and the ways we work on it.

We are the people who file issues and pull requests, attend SIG meetings, Kubernetes meetups, and KubeCon, advocate for its adoption and innovation, run `kubectl get pods`, and contribute in a thousand other vital ways. Read on to learn how you can get involved and become part of this amazing community.

전세계 150개가 넘는 모임, 활발한 활동

왜 쿠버네티스인가



오픈소스

엄청난 인기

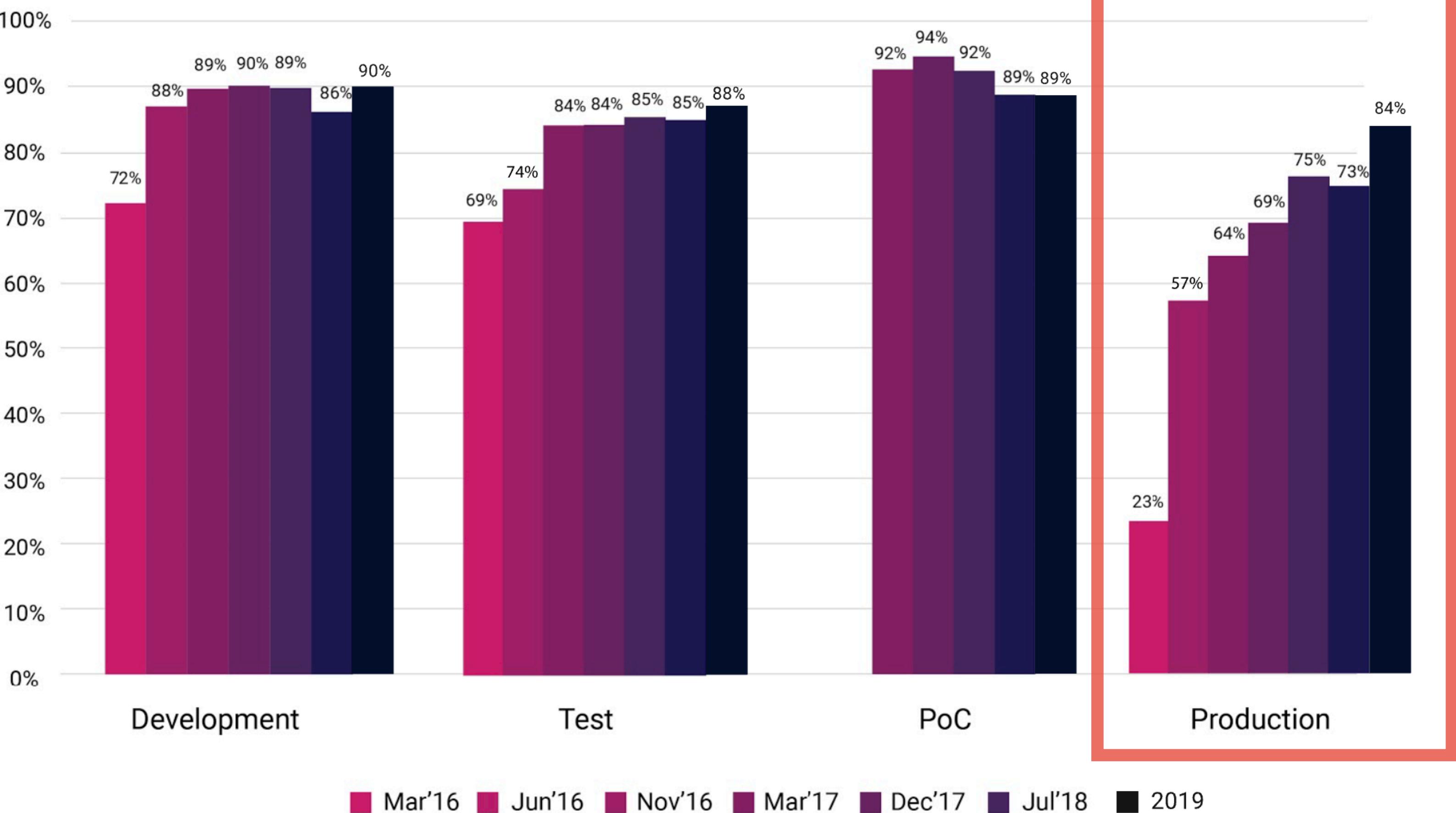
무한한 확장성

사실상의 표준 (de facto)

쿠버네이티스의 인기

2019 CNCF Survey results are here:
Deployments are growing in size and
speed as cloud native adoption
becomes mainstream

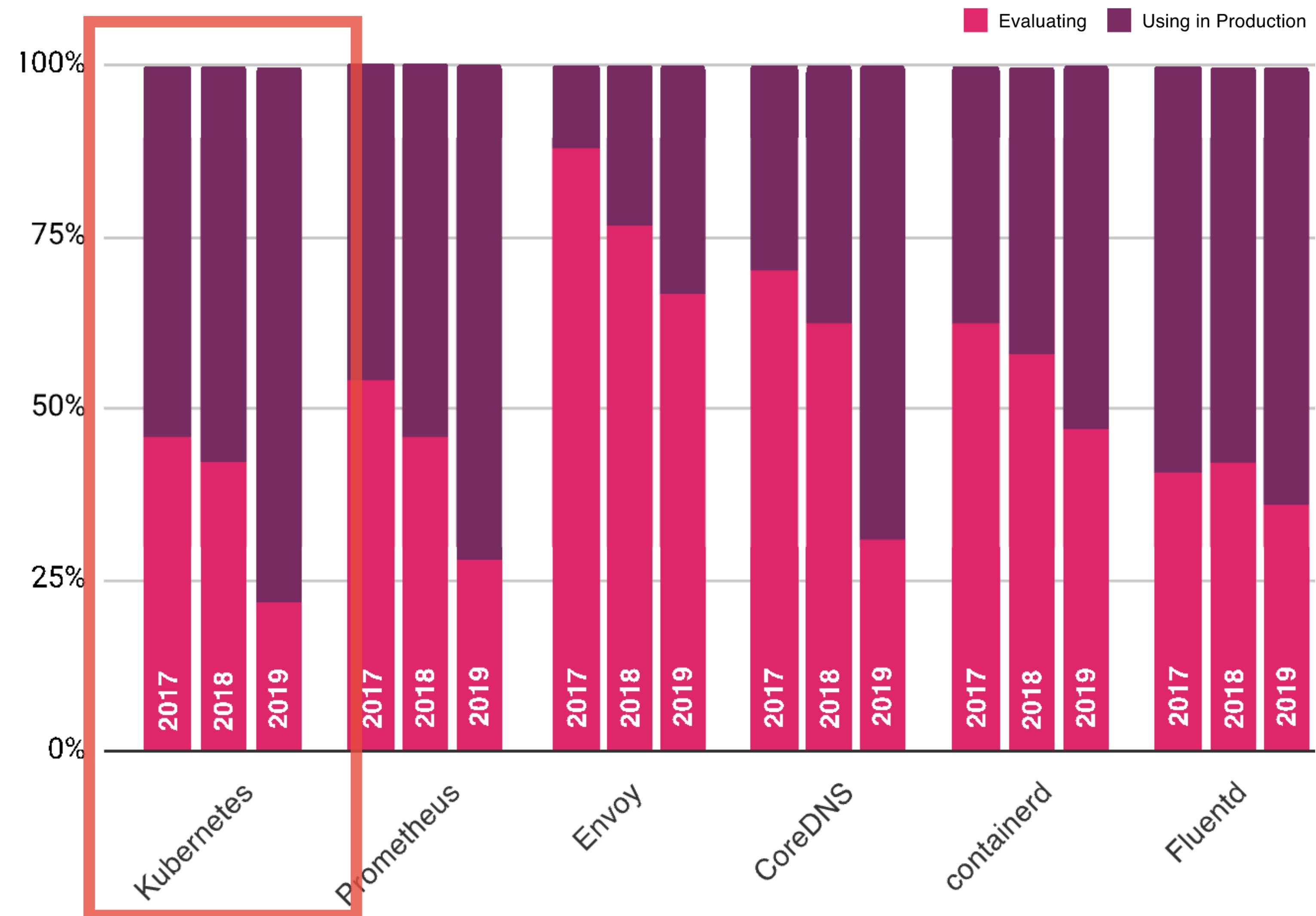
Use of Containers since 2016



쿠버네이티스의 인기

2019 CNCF Survey results are here:
Deployments are growing in size and
speed as cloud native adoption
becomes mainstream

Use of Graduated CNCF Projects



카카오톡 적용 사례를 통해 살펴보는 카카오 클라우드의 Kubernetes as a Service

카카오 프라이빗 클라우드의 KaaS(Kubernetes as a Service) DKOSv3 를 소개 하고, 카카오톡에 적용된 쿠버네티스 사례와 서비스 운영에 대해 공유합니다.

- 서비스 중단 없는 쿠버네티스 기반 컨테이너 클라우드 운영 (dennis.hong)
 - * 카카오의 Kubernetes as a Service, DKOSv3 소개
 - * 사내 인증과 통합된 쿠버네티스 멀티 클러스터 CI/CD 서비스 제공
 - * 자동화된 쿠버네티스 클러스터의 라이프사이클 관리 기능을 통한 무중단 운영
 - * Custom Controller, Custom Resource Definition 을 통한 On-premise 환경에서의 쿠버네티스와 물리 인프라 자원 통합

- 서비스 중단 없는 쿠버네티스 기반 카카오톡 백엔드 배포 및 운영 (greg.47)
 - * 카카오톡 메시징 운영

#클라우드 #쿠버네티스 #k8s #컨테이너 #대규모 #카카오톡 #실서비스 #운영 #사례

대규모 Kubernetes 클러스터 구축기

LINE Developers



Lee Seung | 2019.08.08

Seung is a cloud service developer at LINE.

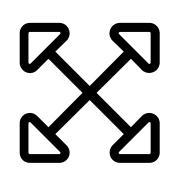
들어가며

LINE에서는 개발자가 개발에만 집중할 수 있도록 개발 외의 인프라 관련 업무를 대신 수행해 주는 Kubernetes 기반 서비스 Nucleo를 자체 개발하여 LINE 내부용으로 사용하고 있습니다. Nucleo에는 현재 800여 개의 앱이 약 4,000개의 pod로 서비스되고 있는데요. 앱의 개수는 계속 빠르게 증가하고 있습니다. 모든 워크로드는 개발 구역(region)에 약 60대, 운영 구역에 약 80대가 설치된 PM(physical machine)과 VM(virtual machine)으로 운영하고 있으며, 그 외에 Nucleo 개발용 클러스터가 약 3~40개, feature 테스트용 노드 3개로 이루어진 미니 클러스터 약 20~30개 가량이 구동되고 있습니다. 이렇게 다수의 Kubernetes 클러스터를 운영하기 위해서는 빠르고 안정적으로 Kubernetes 클러스터를 설치하고 운영할 수 있는 방법이 필요한데요. 그래서 Nucleo 팀에서는 Kubernetes 클러스터 프로비저닝을 목적으로 'Caravan'이라는 도구를 개발하여 사용하고 있습니다. 최근에 팀에서는 이 Caravan이 앞으로 늘어날 앱의 수요에 대응할 수 있는지 알아보기 위해 많은 양의 Kubernetes 클러스터를 생성하고 테스트해 보았는데요. 이번 글에선 이 테스트의 상세한 내용과 테스트를 통해 배운 점에 대해 설명하겠습니다.

왜 쿠버네티스인가

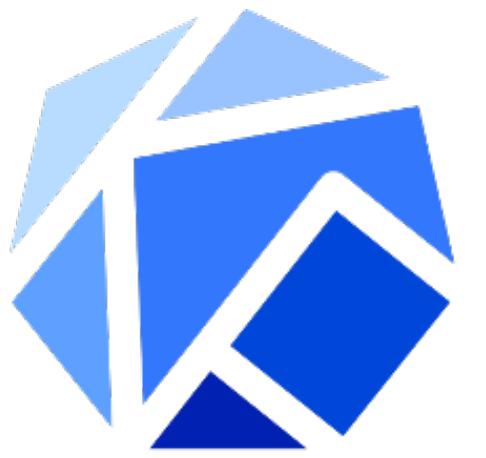
 오픈소스

 엄청난 인기

 무한한 확장성

 사실상의 표준 (de facto)

무한한 확장성

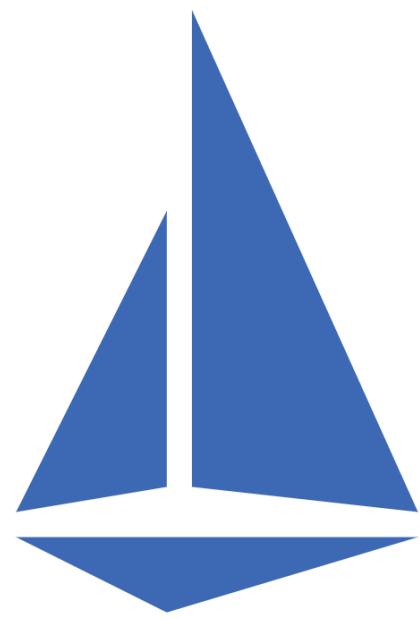


Kubeflow

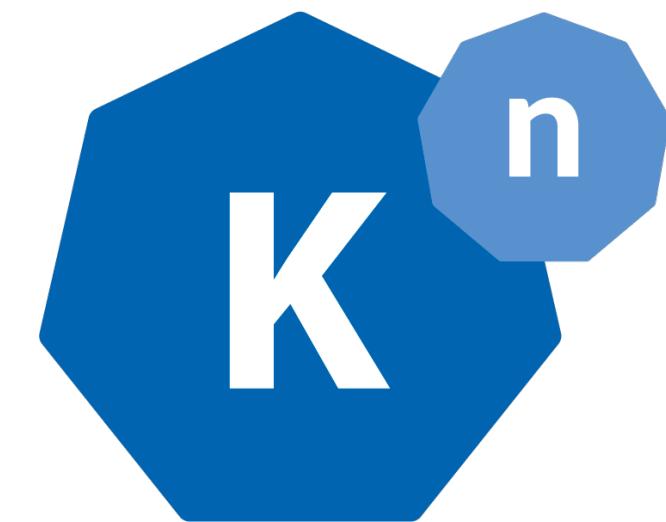
머신러닝



CI/CD



서비스메시



서버리스



가상화 / 개발자 / 데이터센터 / 비즈니스|경제 / 신기술|미래 / 애플리케이션 / 오픈소스 / 운영체제 / 클라우드

2018.11.22

칼럼 | 쏘리! 리눅스, 이제 주인공은 ‘쿠버네티스’다

Matt Asay | InfoWorld

이제 운영체제는 더 이상 중요하지 않다. 이는 개발자나 클라우드에 있어 리눅스가 더 이상 중요하지 않는다는 의미이다.

일어나지 않은 일이 그 증거다. 우분투를 운영하는 캐노니컬은 IBM으로부터 340억 달러의 인수 제안을 받지 않았다. 매각에 관심이 없다는 이야기가 있긴 했지만, 캐노니컬 설립자인 마크 셔틀워스라면 그러한 제안이 왔을 때 받아들였을 것이다.

인수 제안이 캐노니컬에게 가지 않았고, 당분간은 그럴 일이 없다. 이유는 IT산업이 운영체제 자체를 더 이상 가치 있게 생각하지 않기 때문이다. 아니, 차라리 IT산업이 가치 있게 생각하는 새로운 운영체계가 있기 때문이라는 말이 맞겠다. 이는 쿠버네티스라고 불린다.

왜 쿠버네티스인가

 오픈소스

 엄청난 인기

 무한한 확장성

 사실상의 표준 (de facto)

데 팩토

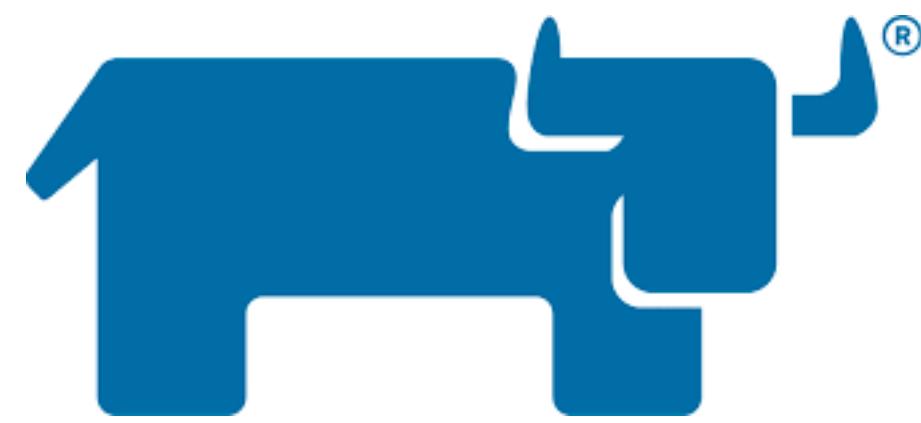
법률 등의 명시적인 방법으로 인정되진 않았지만 다들 암묵적으로 공식처럼 여기는 것

文A

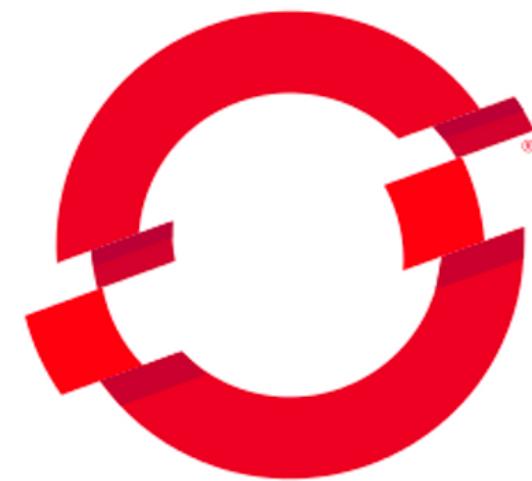


데 팩토([라틴어](#): De facto)는 사실상의 의미로 쓰이는 표현으로, 법적으로 공인된 사항이 아니더라도 실제 존재하는 사례를 가리키는 말이다.)이라 부르기도 한다.

Kubernetes Native Platform



Rancher (by SUSE)



Red Hat OpenShift (by IBM)



Tanzu (by VMware)

사실상의 표준



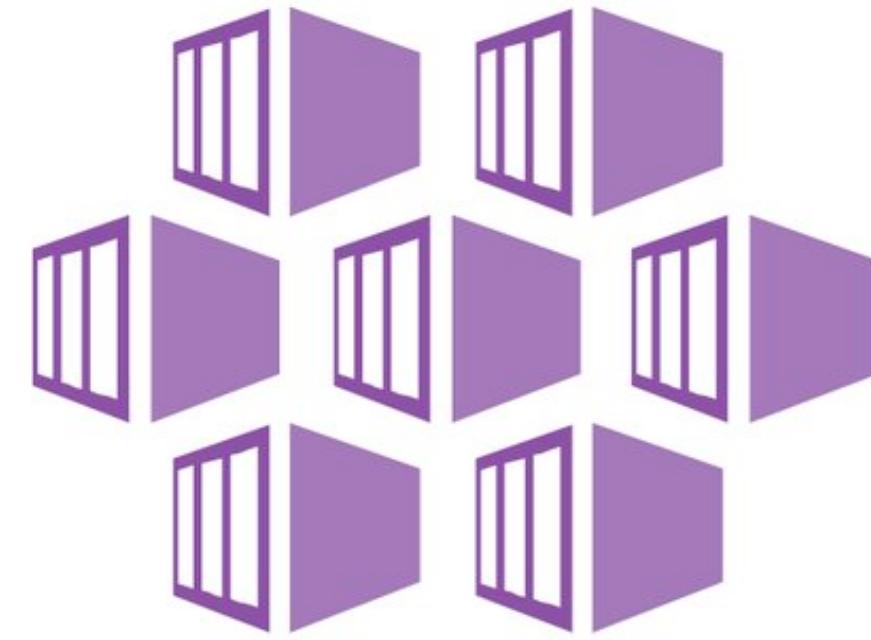
도커도 어쩔수없이 쿠버네티스를 지원!

사실상의 표준



EKS

Amazon
Elastic Kubernetes Service



AKS

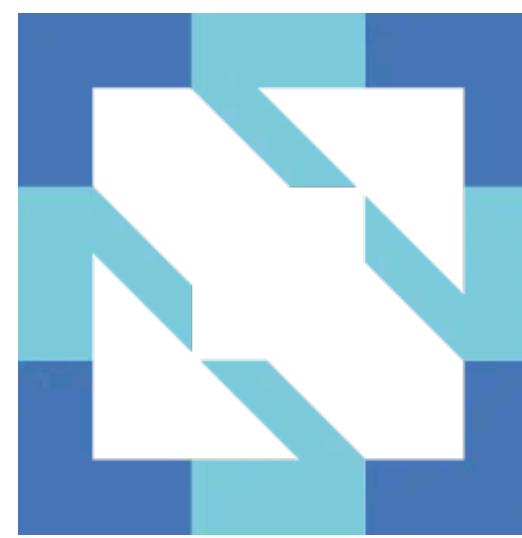
Azure
Kubernetes Service



GKE

Google
Kubernetes Engine

CNCF



CLOUD NATIVE COMPUTING FOUNDATION



클라우드 환경에 적합한 컴퓨팅 기술을 지원

컨테이너

CI/CD

오페스트레이션

모니터링

서비스 디스커버리

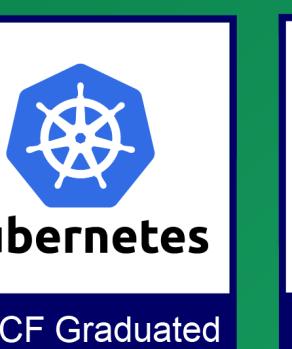
1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices



3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: cncf.io/ck
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application



CNCF Graduated

CNCF Graduated

5. SERVICE PROXY, DISCOVERY, & MESH

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing



2. CI/CD

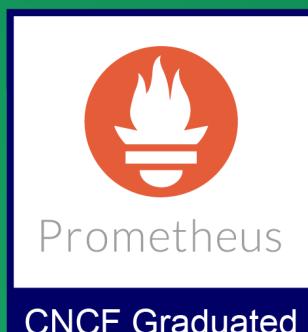
- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing
- Argo is a set of Kubernetes-native tools for deploying and running jobs, applications, workflows, and events using GitOps paradigms such as continuous and progressive delivery and MLOps



CNCF Incubating

4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger



CNCF Graduated



CNCF Graduated



CNCF Graduated



CNCF Incubating



CNCF

컨테이너

CI/CD

오케스트레이션

모니터링

서비스 디스커버리

네트워크 & 보안

분산 디비 & 저장소

메시징

컨테이너 런타임

배포 (인증)



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.



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.



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.



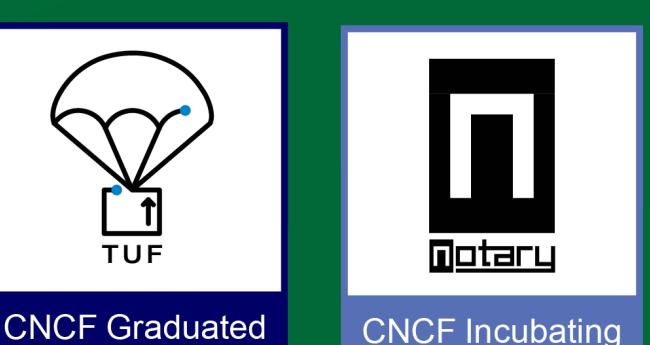
8. STREAMING & MESSAGING

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.



10. SOFTWARE DISTRIBUTION

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



사실상의 표준 (de facto)

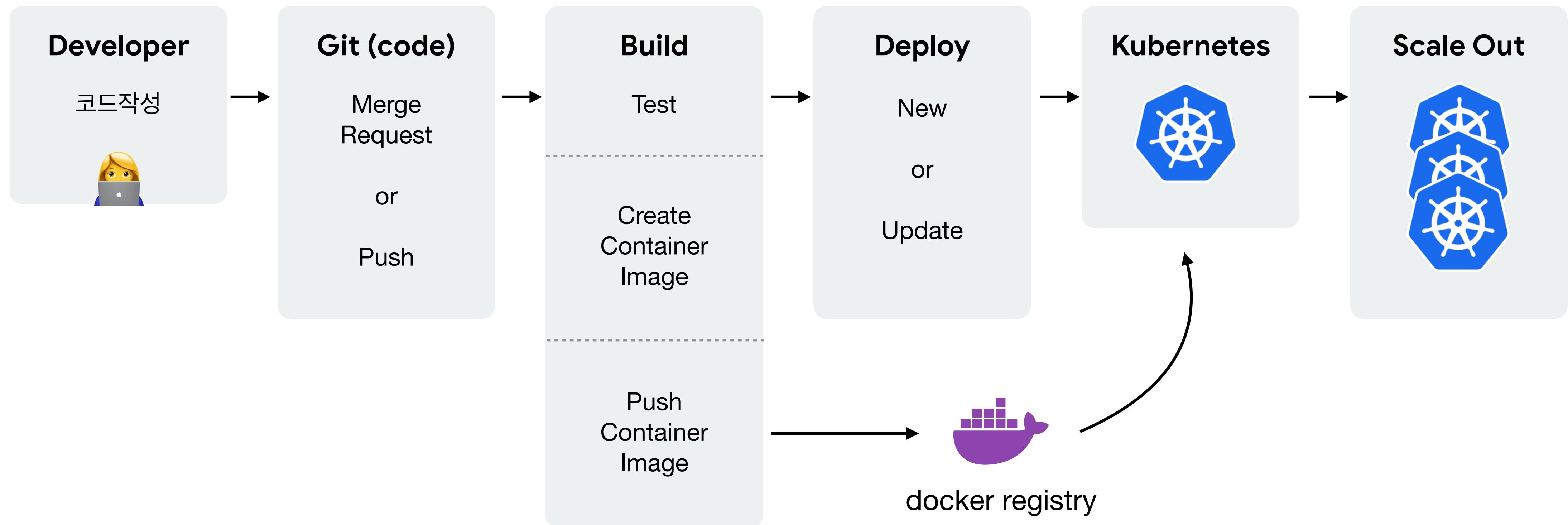


컨테이너 오케스트레이션의 사실상의 표준 (de facto)
Cloud Native의 핵심역할

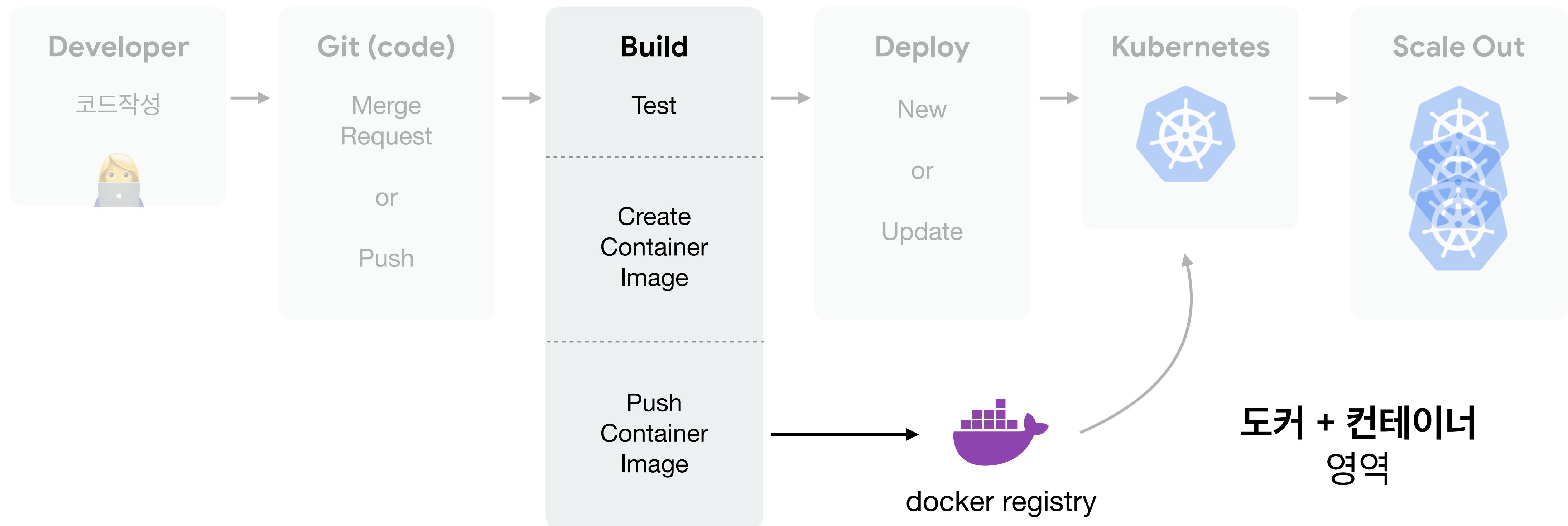
쿠버네티스

어떤걸 배울까?

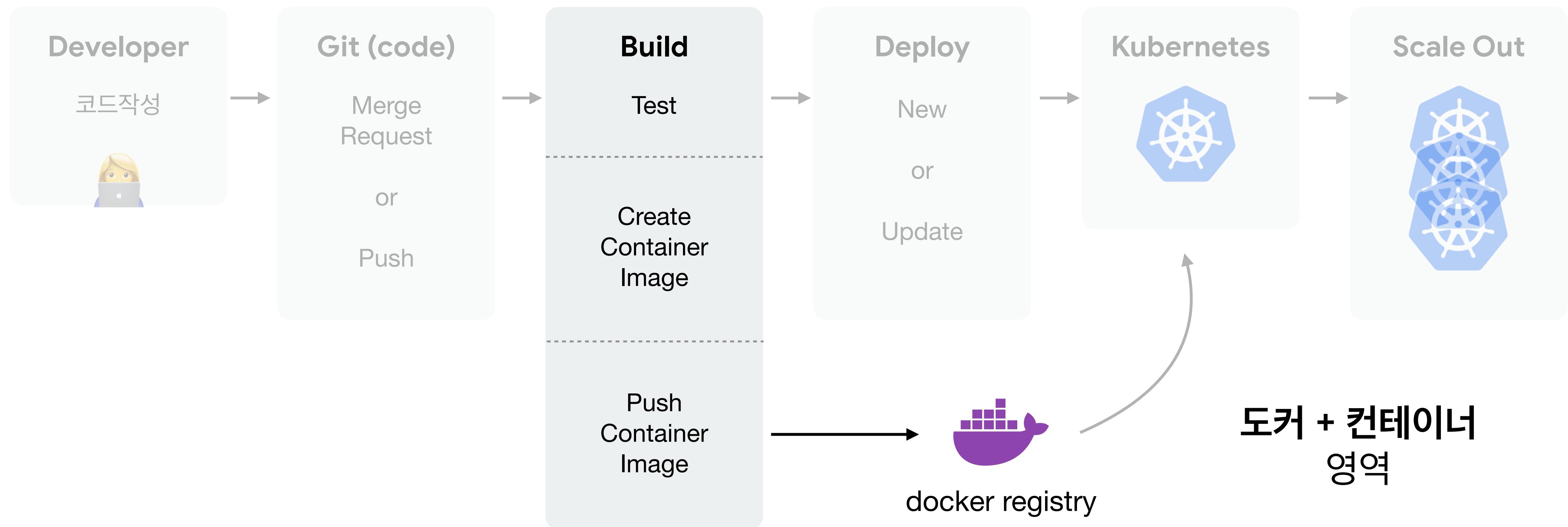
어떤걸 배울까?



어떤걸 배울까?

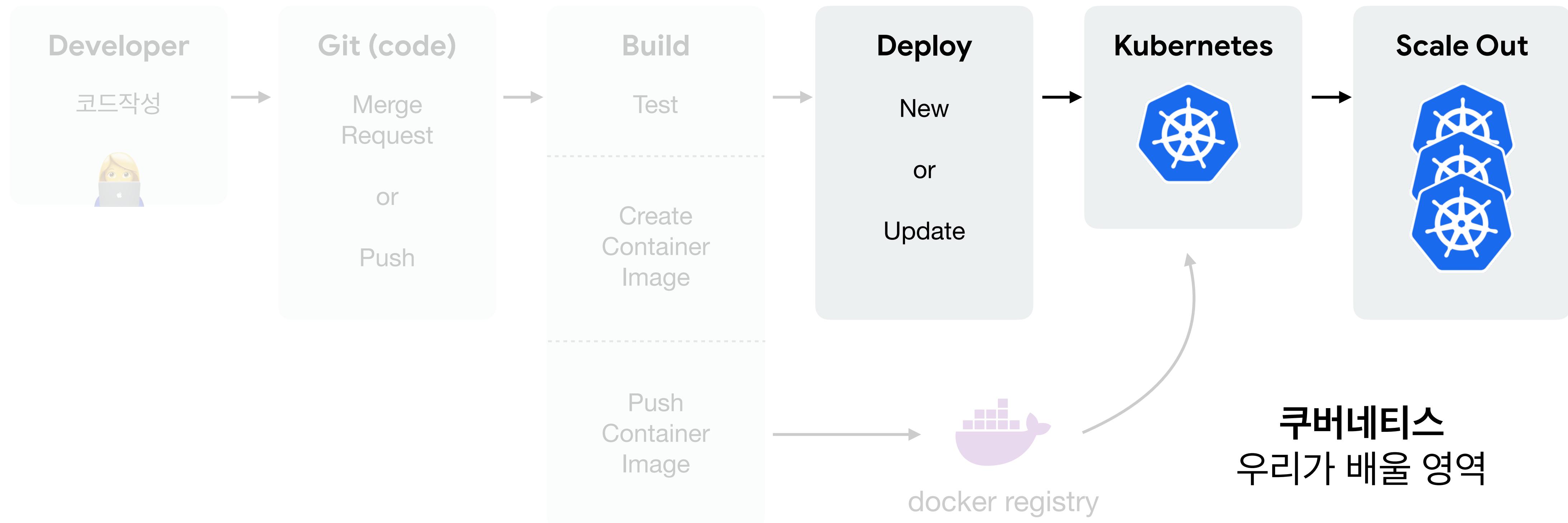


어떤걸 배울까?

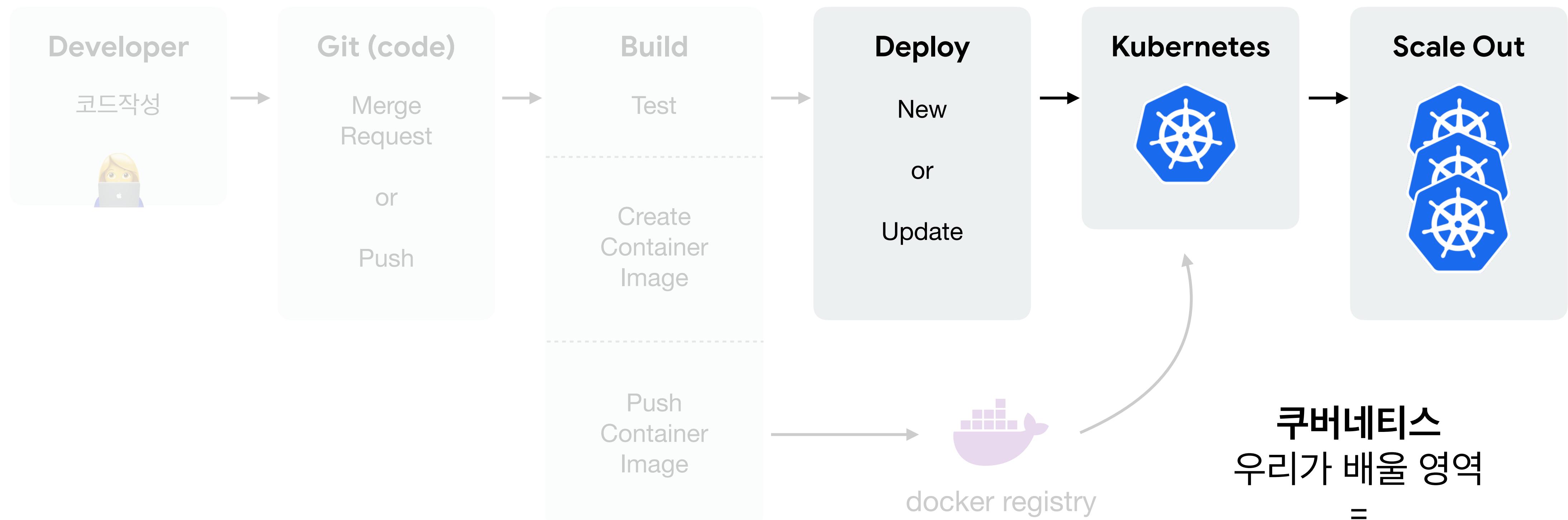


도커를 모른다면 쿠버네티스를 완벽하게 이해할 수 없습니다

어떤걸 배울까?



어떤걸 배울까?



어떤걸 배울까?

학습범위

도커 컨테이너 실행하기

쿠버네티스에 컨테이너 배포하기

외부 접속 설정 하기

스케일 아웃 하기

그외 고급기능 소개

어떤걸 배울까?

학습범위

도커 컨테이너 실행하기

쿠버네티스에 컨테이너 배포하기

외부 접속 설정 하기

스케일 아웃 하기

그외 고급기능 소개

도커와 도커컴포즈를 이용한
멀티 컨테이너 관리

어떤걸 배울까?

학습범위

도커 컨테이너 실행하기

쿠버네티스에 컨테이너 배포하기

외부 접속 설정 하기

스케일 아웃 하기

그외 고급기능 소개

실습(hands-on) 환경 만들기

kubectl 사용법

pod, deployment, service 등

기본 리소스 학습

어떤걸 배울까?

학습범위

도커 컨테이너 실행하기

쿠버네티스에 컨테이너 배포하기

외부 접속 설정 하기

스케일 아웃 하기

그외 고급기능 소개

Cluster IP, NodePort,
LoadBalancer, Ingress
서비스 타입 학습
서비스 디스커버리 학습

어떤걸 배울까?

학습범위

도커 컨테이너 실행하기

쿠버네티스에 컨테이너 배포하기

외부 접속 설정 하기

스케일 아웃 하기

그외 고급기능 소개

부하에 따른 컨테이너 개수 조정

최소 리소스 요청 설정

오토스케일링

어떤걸 배울까?

학습범위

도커 컨테이너 실행하기

쿠버네티스에 컨테이너 배포하기

외부 접속 설정 하기

스케일 아웃 하기

그외 고급기능 소개

HELM 패키지 매니저 소개
GitOps, ServiceMesh 소개

어떤걸 배울까?

다루지 않는 범위

다양한 환경별 특징 (bare metal, EKS, ...)

쿠버네티스 패턴 (사이드카, 어댑터, ...)

관련 생태계 (서비스메시, 서비스, ...)

GitOps CI/CD

승인제어 등 고급 기능

어떤걸 배울까?

쿠버네티스 = 어렵다

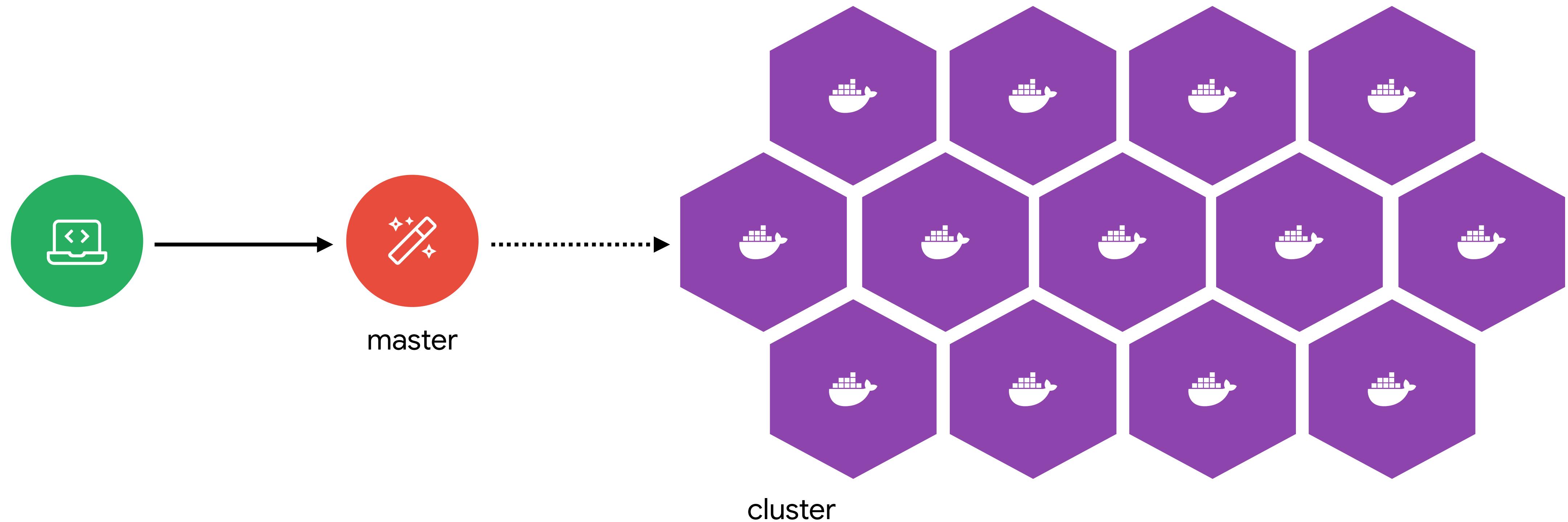
어떤걸 배울까?

쿠버네티스 = 어렵다 (모든 것을 다 하려면)

어떤걸 배울까?

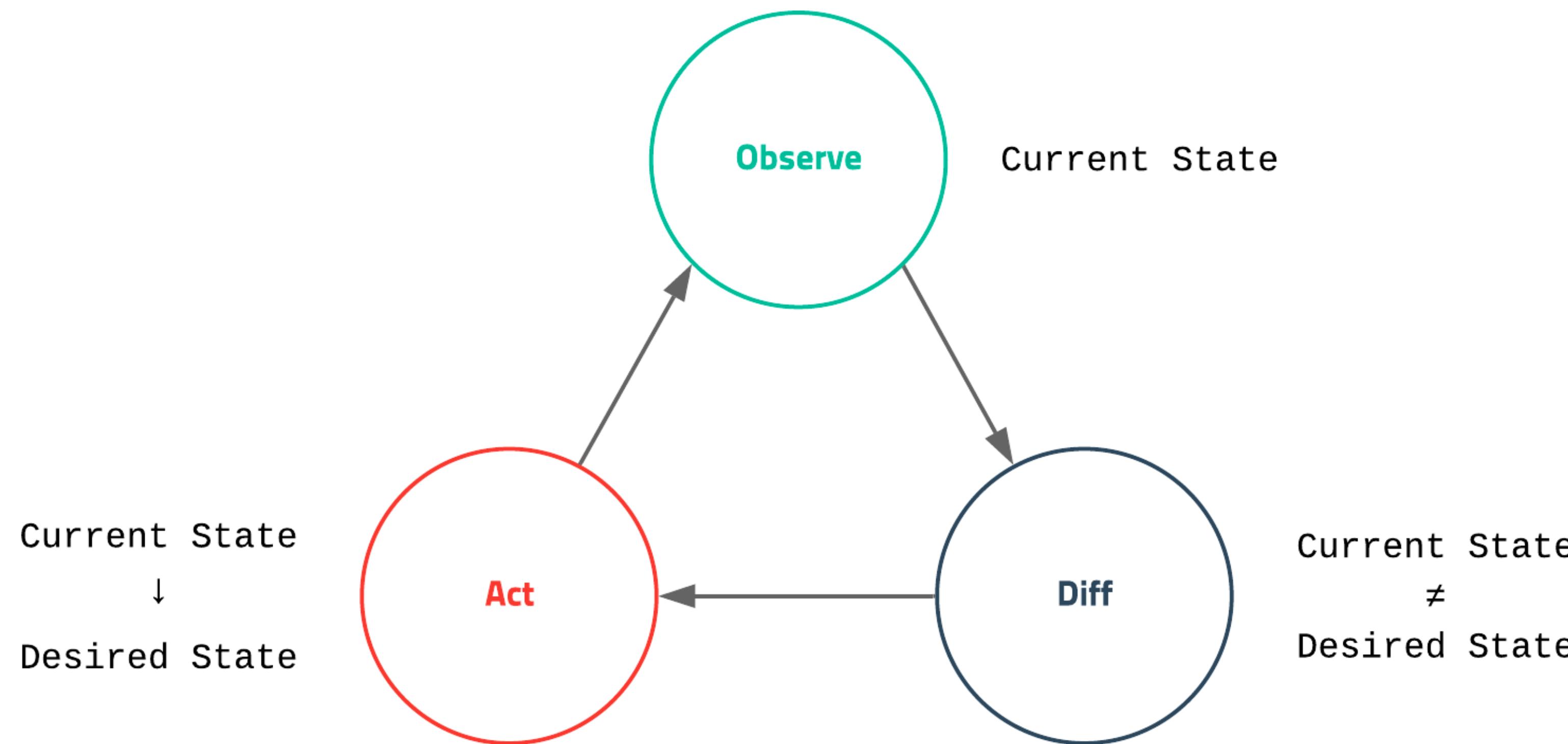
구성요소를 이해하고
동작원리를 파악하고
기본적인 사용법을 익혀봅니다

어떻게 배울까?



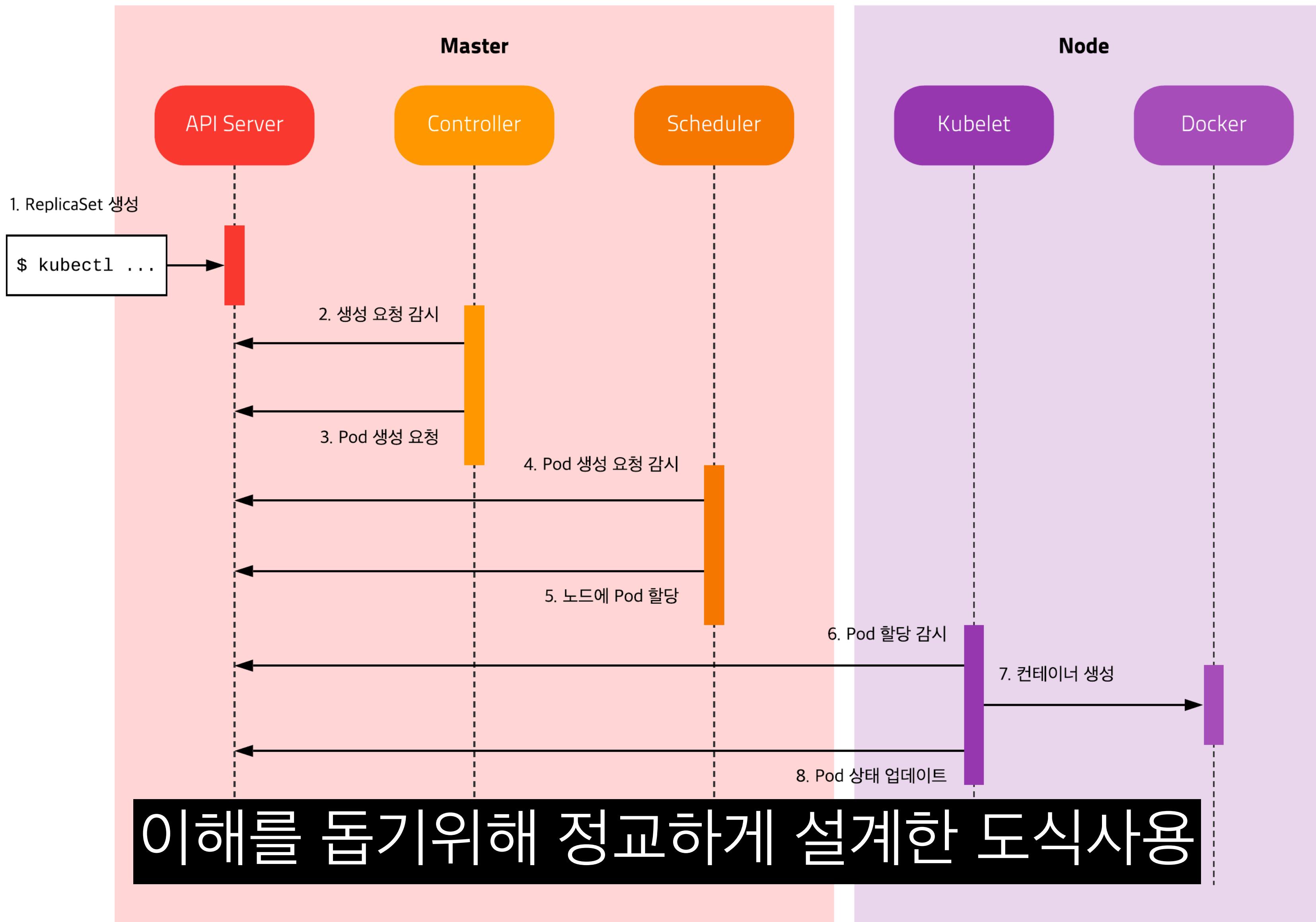
이해를 돋기위해 정교하게 설계한 도식사용

어떻게 배울까?



이해를 돋기위해 정교하게 설계한 도식사용

어떻게 배울까?



어떻게 배울까?

```
kubectl run whoami --image subicura/whoami:1 # deprecated soon..
kubectl get po
kubectl get pod
kubectl get pods
kubectl get pods -o wide
kubectl get pods -o yaml
kubectl get pods -o json
kubectl logs whoami-<xxxx>
kubectl logs -f whoami-<xxxx>
kubectl exec -it whoami-<xxxx> sh
kubectl describe pods whoami-<xxxx>
kubectl delete pods whoami-<xxxx>
kubectl get pods
kubectl get all
kubectl delete deployment/whoami
```

따라하면서 자연스럽게 익힐 수 있는 실습 (hands-on)

어떻게 배울까?

Exam 1. 다음 조건을 만족하는 pod을 만들어 보세요.

- Name: mongodb
- Labels: app => mongo
- Container Name: mongodb
- Image: mongo:4

개인실습 자료제공



Thanks!