

쿠버네티스 자동화 환경 서비스

Camucamu

윤순상 유성욱 김태교 유태균 권재성

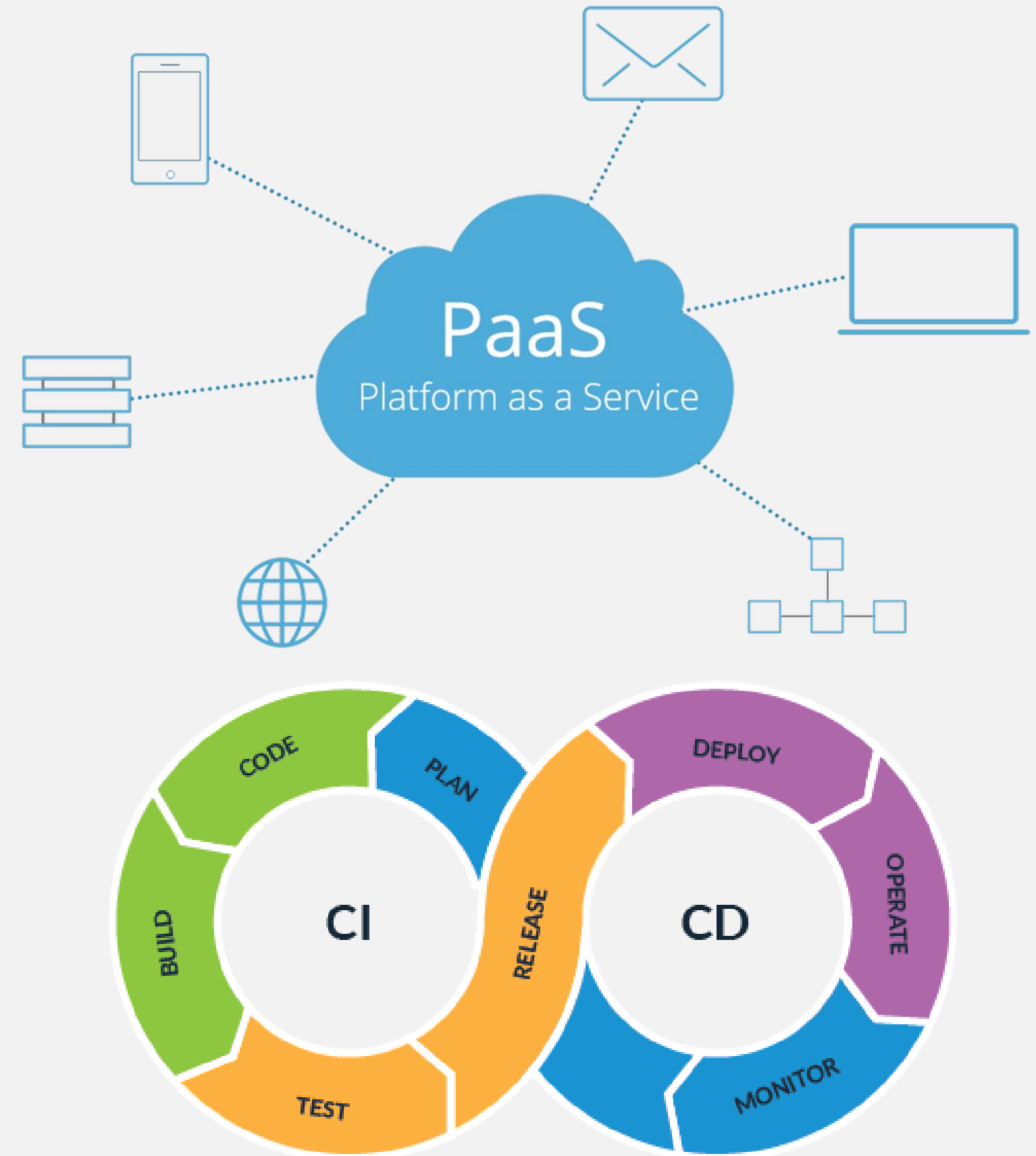
2024.04.04

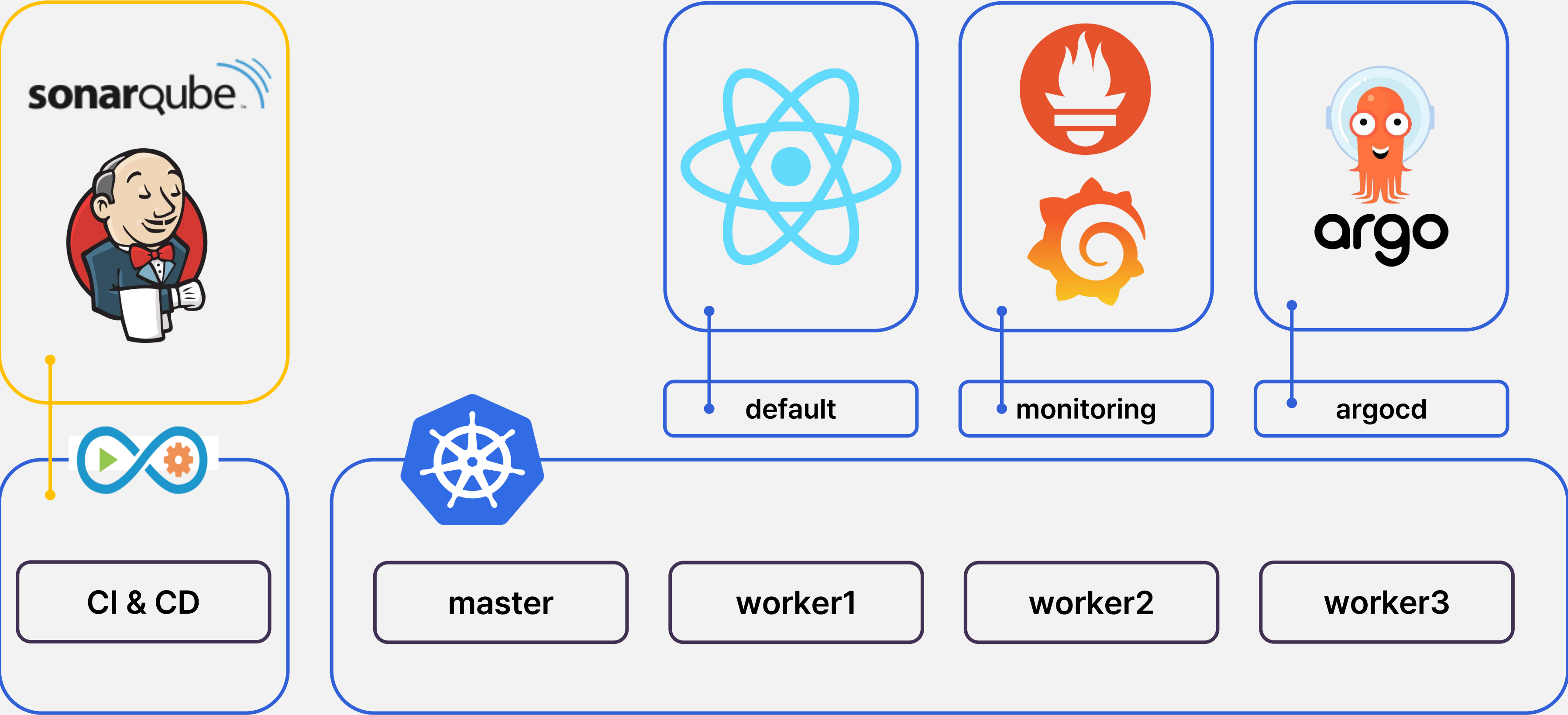
RAPA KAKAO

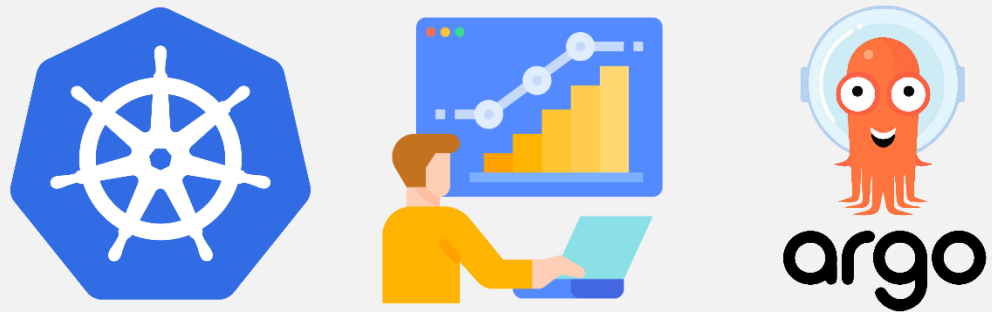
Engineer 4

쿠버네티스 자동화 환경 배포 서비스

- ✓ 사용자가 원하는 구성에 따라 쿠버네티스 클러스터를 **자동으로 배포**하고 구성할 수 있는 기능을 제공하며 개발자는 간편한 인터페이스를 통해 애플리케이션을 배포하고 관리할 수 있습니다. 또한 배포 환경을 구성하고 및 롤백 기능을 통해 **안정적인 배포**를 보장합니다.
- ✓ 쿠버네티스 클러스터와 애플리케이션의 상태를 지속적으로 **모니터링**하고 문제가 발생할 경우 즉시 경고를 제공하여 **신속한 대응**을 가능하게 합니다.
- ✓ 자동화된 프로비저닝 및 배포 프로세스로 개발 및 운영 팀은 시간과 노력을 절약하고 생산성을 향상시킬 수 있으며 일관된 환경에서 애플리케이션을 운영함으로써 **안정성과 신뢰성을 확보**하고 장애 상황에 대비할 수 있습니다.







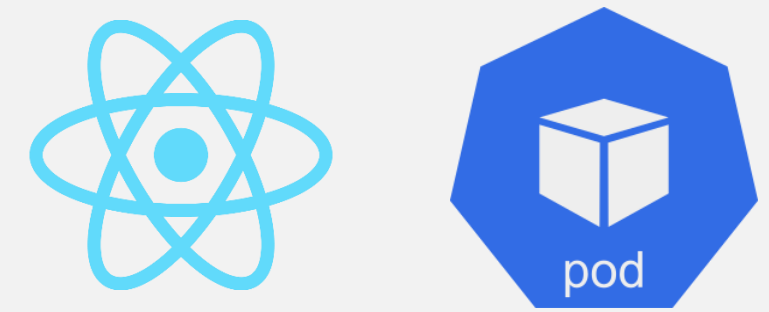
Kubernetes

k8s Cluster → 1.28v
Namespace → metallb
Namespace → argocd
Namespace → monitoring



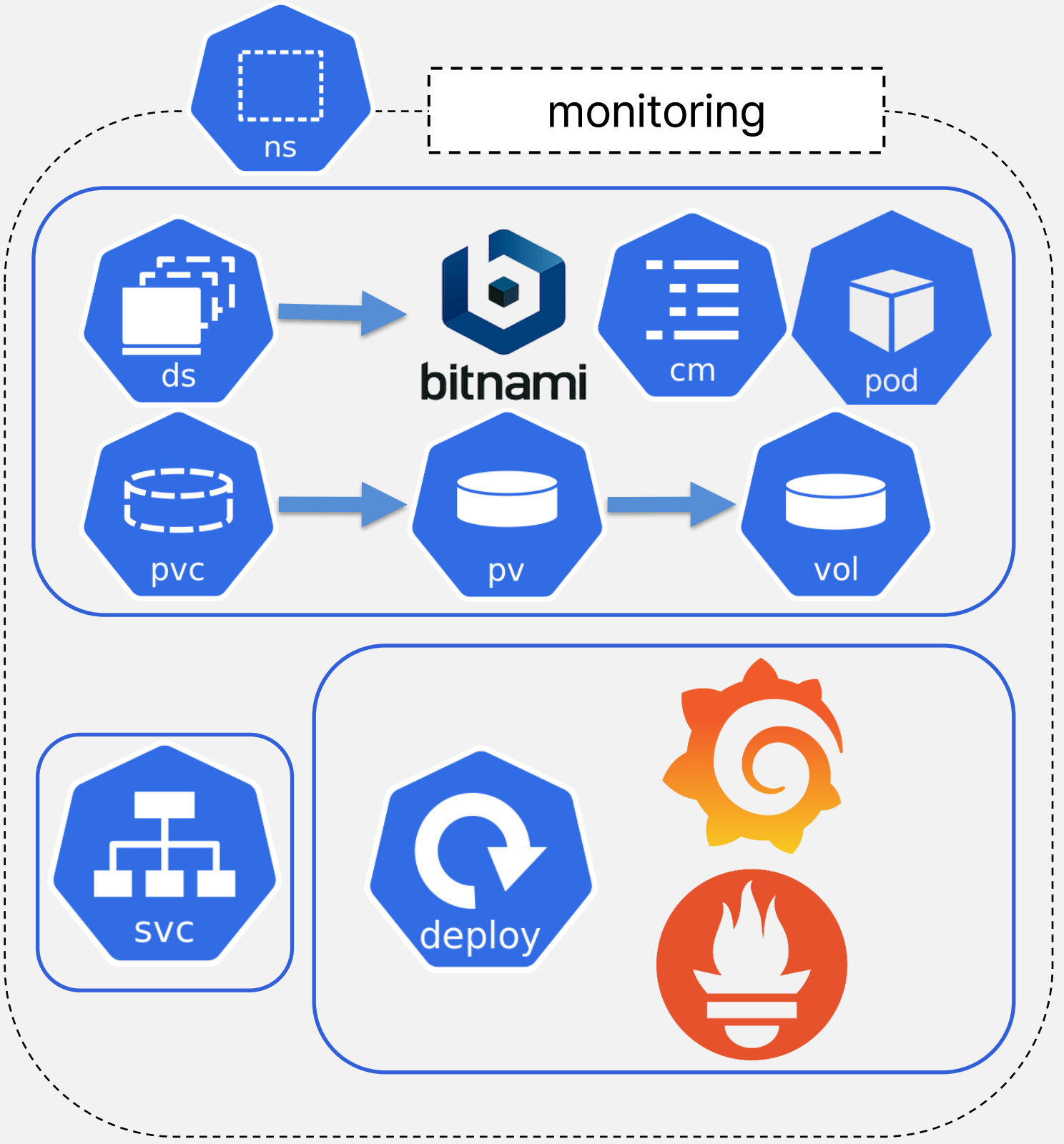
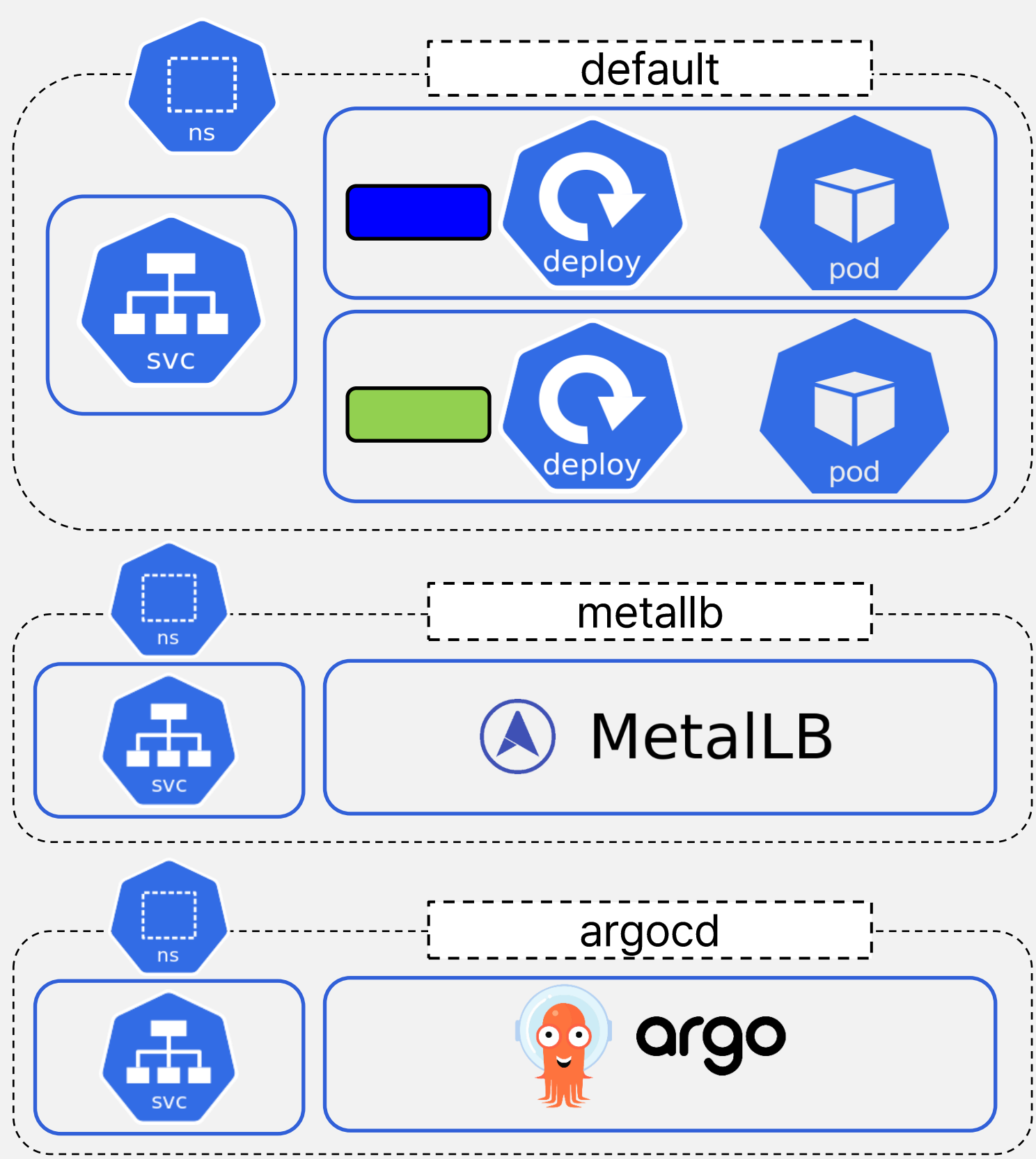
CI & CD

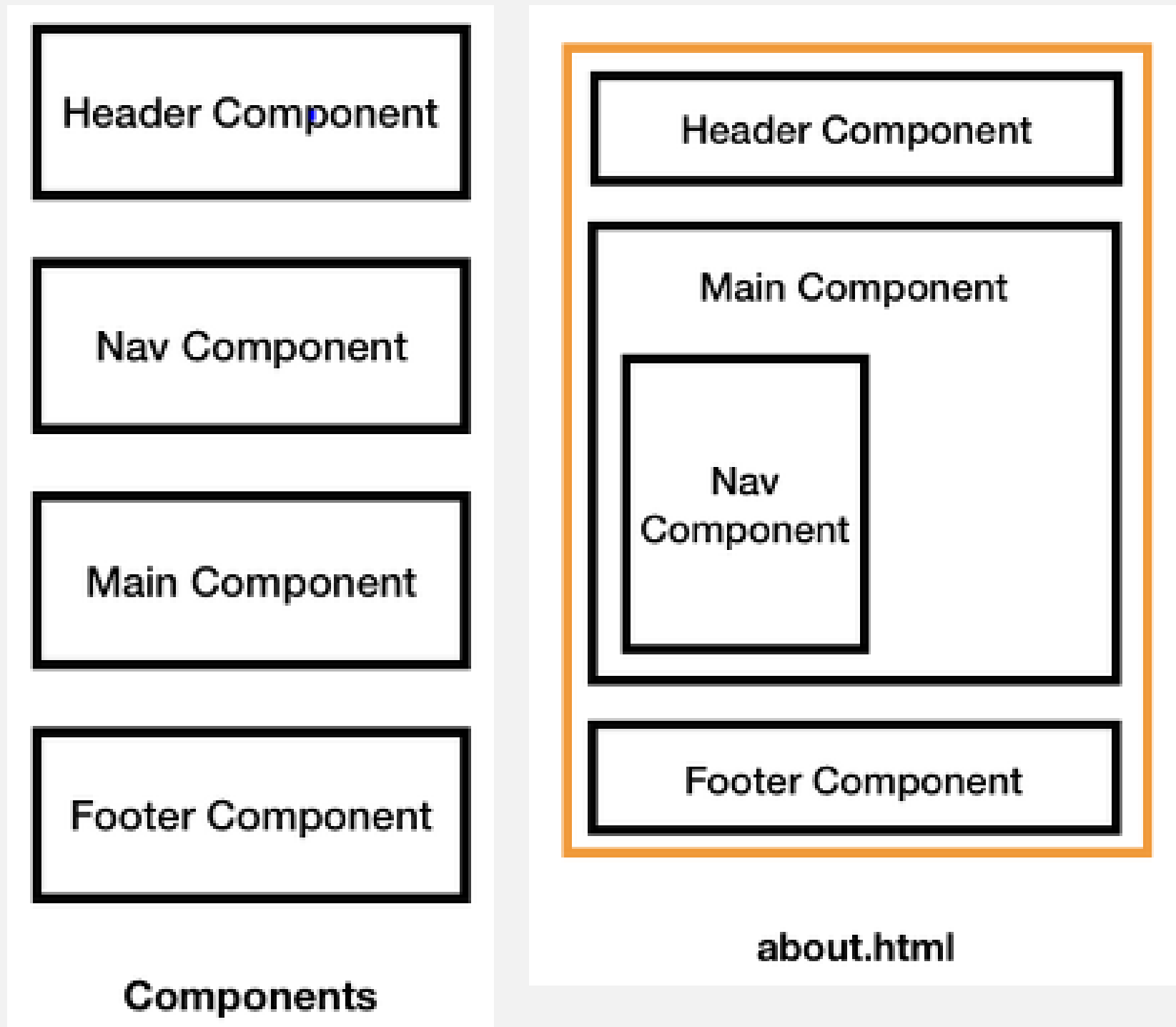
Jenkins & Sonarqube
Git Webhooks
CI Pipeline



Application

Language) HTML, CSS, JS
Framework) React
Build) Dockerfile





컴포넌트 사용

React는 화면을 여러 컴포넌트로 쪼개서 구성한 페이지 내에서 각 부분을 **독립된 컴포넌트**로 만들고, 이 컴포넌트를 조립해 화면을 구성

- ✓ 컴포넌트 단위로 쪼개져 있어 코드 파악이 쉽다.
- ✓ 기능 단위, UI 단위로 캡슐화 시켜 코드를 관리하기 때문에 **재사용성**이 높다.

코드를 반복해 입력할 필요 없이 컴포넌트만 import하여 사용한다는 간편함이 있으며, 애플리케이션이 복잡하더라도 **코드의 유지보수** 및 관리에 효율적

React-Route-Dom 방식 이용

```
function App() {  
  return  
    <BrowserRouter>  
      <h1>my web site</h1>  
      <ul>  
        <li><a href="/">home</a></li>  
        <li><a href="/product">product</a></li>  
        <li><a href="/about">about</a></li>  
      </ul>  
      <Routes>  
        <Route path="/" element={<>home page</>></Route>  
        <Route path="/product" element={<>Product page</>></Route>  
        <Route path="/about" element={<>About page</>></Route>  
      </Routes>  
    </BrowserRouter>  
}
```

a태그를 link 컴포넌트로
바꿔주면 페이지의 로드 없이
about 페이지 내용으로 변경


```
function App() {  
  return  
    <BrowserRouter>  
      <h1>my web site</h1>  
      <ul>  
        <li><Link to="/">home</Link></li>  
        <li><Link to="/product">product</Link></li>  
        <li><Link to="/about">about</Link></li>  
      </ul>  
      <Routes>  
        <Route path="/" element={<>home page</>></Route>  
        <Route path="/product" element={<>Product page</>></Route>  
        <Route path="/about" element={<>About page</>></Route>  
      </Routes>  
    </BrowserRouter>  
}
```

URL의 패스가 달라지면 패스가
일치하는 컴퍼넌트가 렌더
링 되게 된다.

Single
Page
Application

Team Camucamu

HomeAboutBlog



토이프로젝트-Camucamu

팀장 : 윤순상
팀원 : 김태고 권재성 유태균 유성욱
파이팅!!

Read Blog

TeamCamucamu@2024

Team Camucamu

HomeAboutBlog

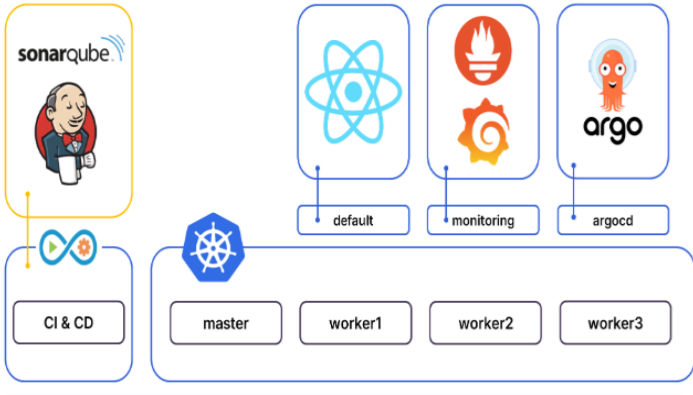
쿠버네티스 자동화 배포 서비스

사용자가 원하는 구성에 따라 쿠버네티스 클러스터를 자동으로 배포하고 구성할 수 있는 기능을 제공하며 개발자는 간편한 인터페이스를 통해 애플리케이션을 배포하고 관리할 수 있습니다. 또한 롤링 업데이트 및 롤백 기능을 통해 안정적인 배포를 보장합니다. 쿠버네티스 클러스터와 애플리케이션의 상태를 지속적으로 모니터링하고 문제가 발생할 경우 즉시 경고를 제공하여 신속한 대응을 가능하게 합니다. 자동화된 프로비저닝 및 배포 프로세스로 개발 및 운영 팀은 시간과 노력을 절약하고 생산성을 향상시킬 수 있으며 일관된 환경에서 애플리케이션을 운영함으로써 안정성과 신뢰성을 확보하고 장애 상황에 대비할 수 있습니다.

사용도구

- K8S
- Docker
- SonarQube
- ArgoCD
- Jenkins
- Grafana
- ReactJS
- NodaJS

인프라 구성



TeamCamucamu@2024

Team Camucamu

HomeAboutBlog

쿠버네티스 자동화 배포 서비스 도구

K8S

Read More

Docker

Read More

SonarQube

Read More

ArgoCD

Read More

Jenkins

Read More

Prometheus

Read More

Javascript

Read More

Bootstrap 5

Read More

React JS

Read More

Node JS

Read More

TeamCamucamu@2024

Team Camucamu

HomeAboutBlog

쿠버네티스

Book by Brian Kernighan and Dennis Ritchie

Back




K8S

안쿠버네티스(k8s 또는 큐브 라고도 함)는 컨테이너화된 애플리케이션을 배포, 관리, 확장할 때 수반되는 다수의 수동 프로세스를 자동화하는 오픈소스 컨테이너 오케스트레이션 플랫폼입니다. 원래 Google의 엔지니어 팀에서 Borg 프로젝트로 개발하고 설계한 쿠버네티스는 2015년에 CNCF(Cloud Native Computing Foundation)에 기부되었습니다. Red Hat은 쿠버네티스가 출시되기 전부터 Google과 협력하여 개발에 참여한 초기 기업 중 하나였으며 쿠버네티스 업스트림 프로젝트에서 두 번째 주요 기여자가 되었습니다.

TeamCamucamu@2024

Team Camucamu

HomeAboutBlog



토크 프로젝트-Camucamu

팀장 : 윤순상

팀원 : 김태교 권재성 유태균 유성욱


파이팅!!

Read Blog

TeamCamucamu@2024

Team Camucamu

HomeAboutBlog



토크 프로젝트-Camucamu

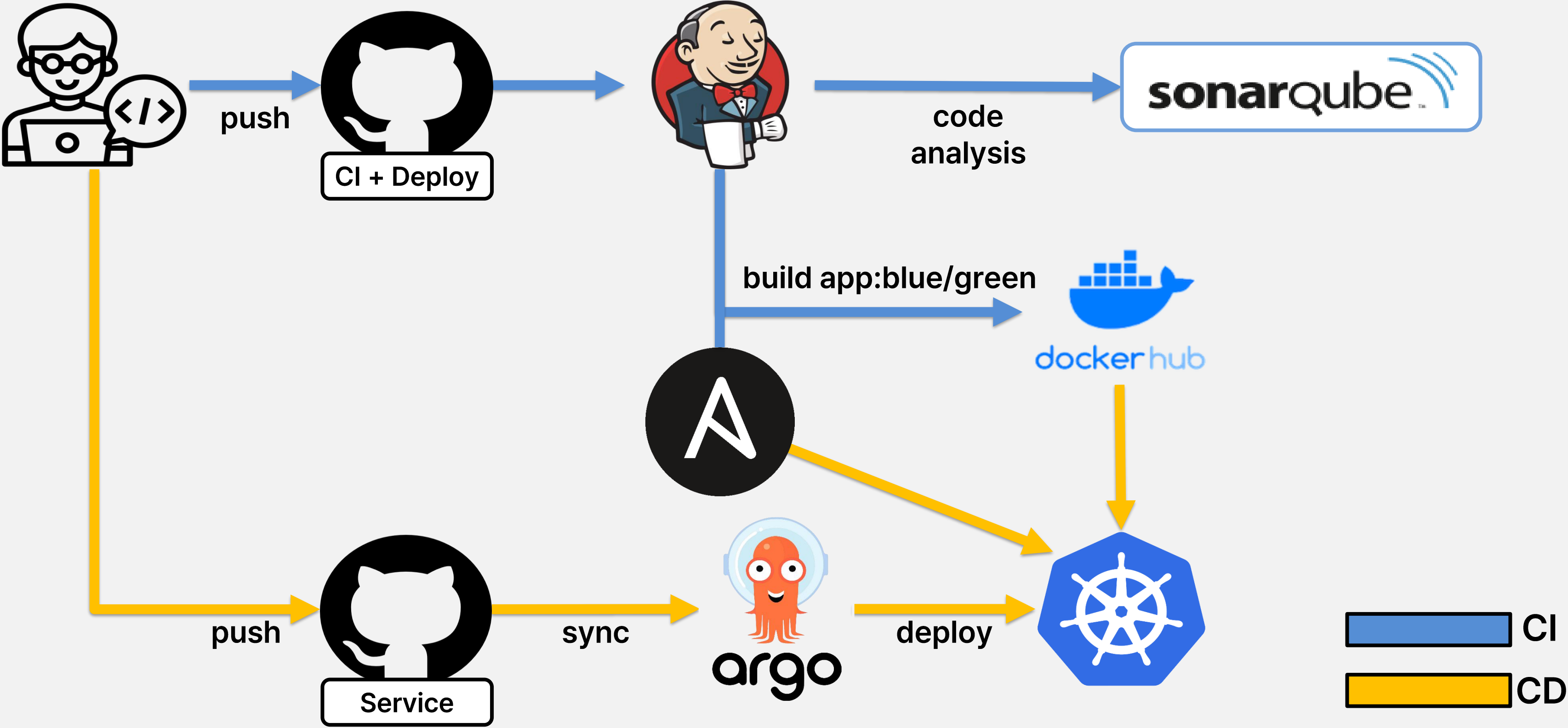
팀장 : 윤순상

팀원 : 김태교 권재성 유태균 유성욱

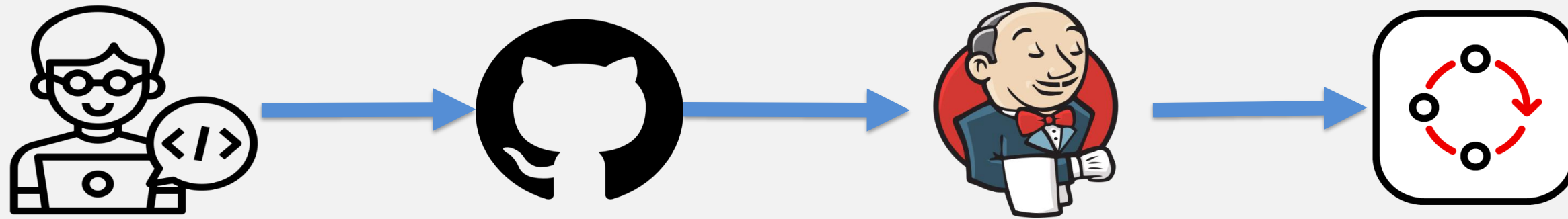
파이팅!!

Read Blog

TeamCamucamu@2024



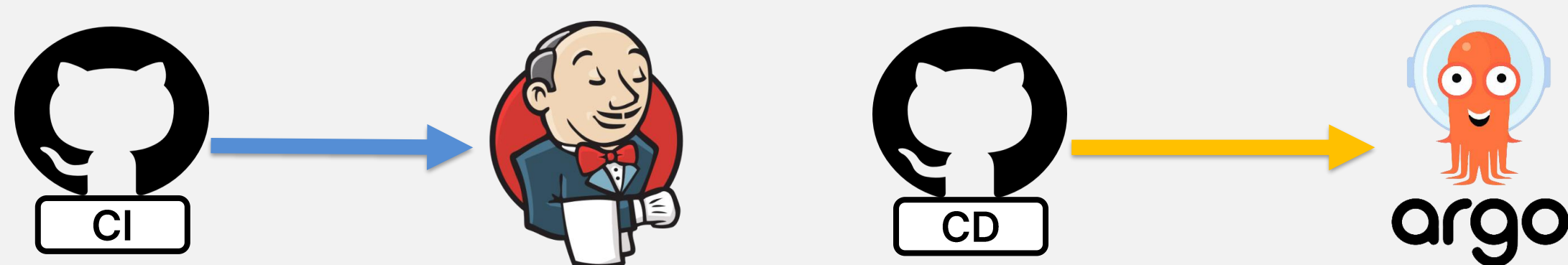
- ✓ CI 자동화 → 깃허브의 커밋 정보를 보고 파이프라인이 실행되도록 자동화 작업이 필요

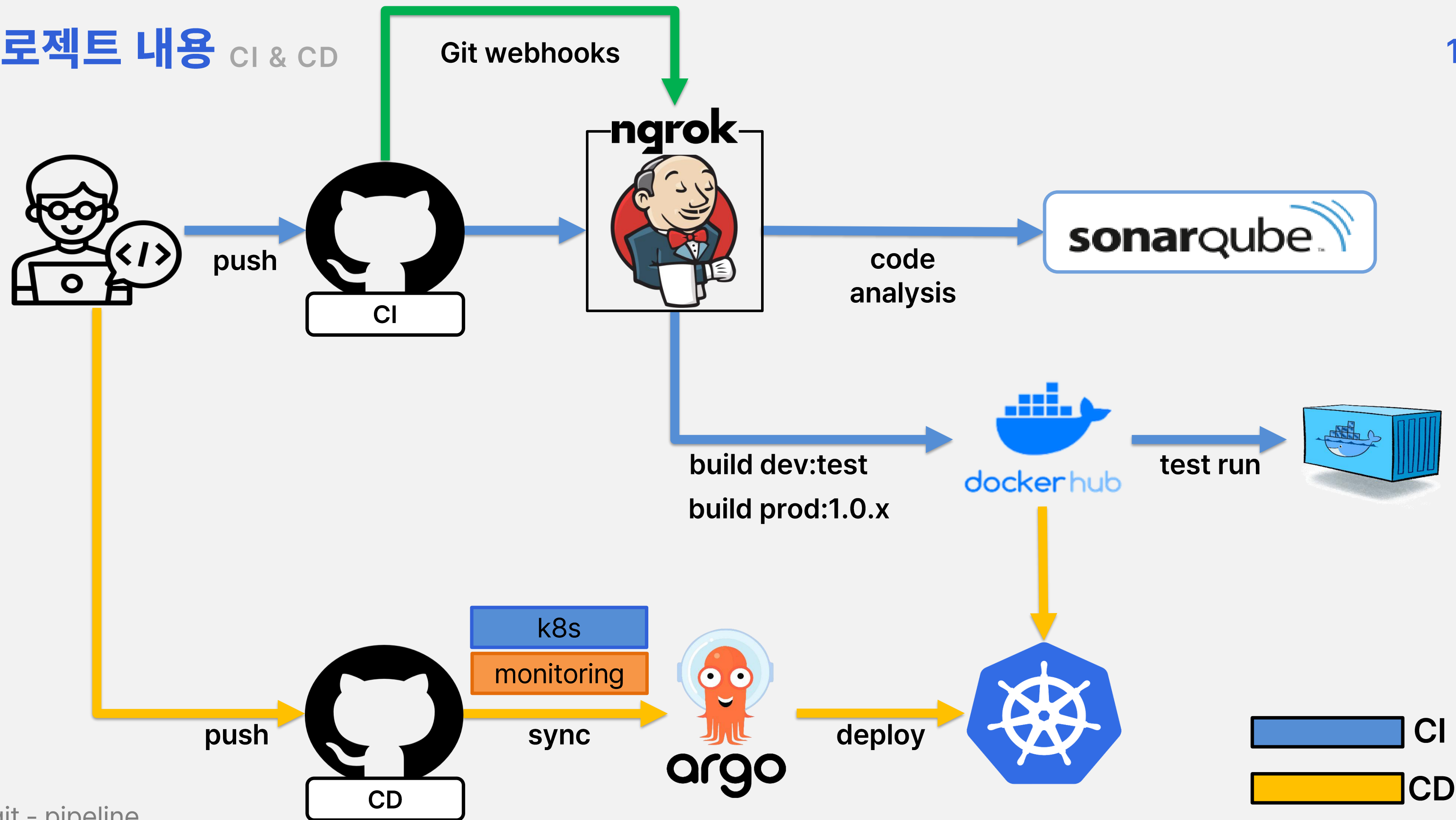


- ✓ 컨테이너 테스트 제공 → 테스트 이미지와 배포 이미지의 구분



- ✓ CI & CD의 구분 → 모호한 자동화 작업을 확실하게 분리





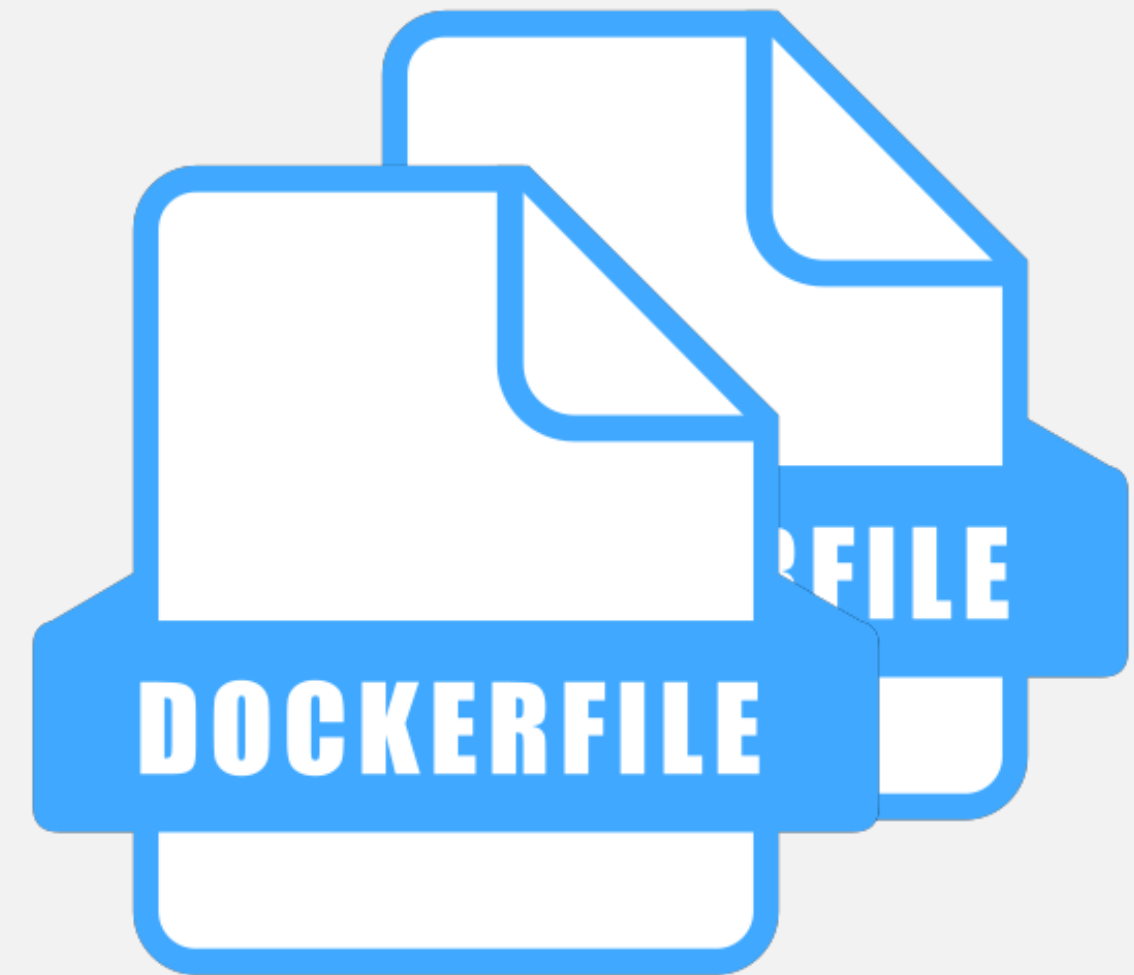
애플리케이션 컨테이너로 만들기

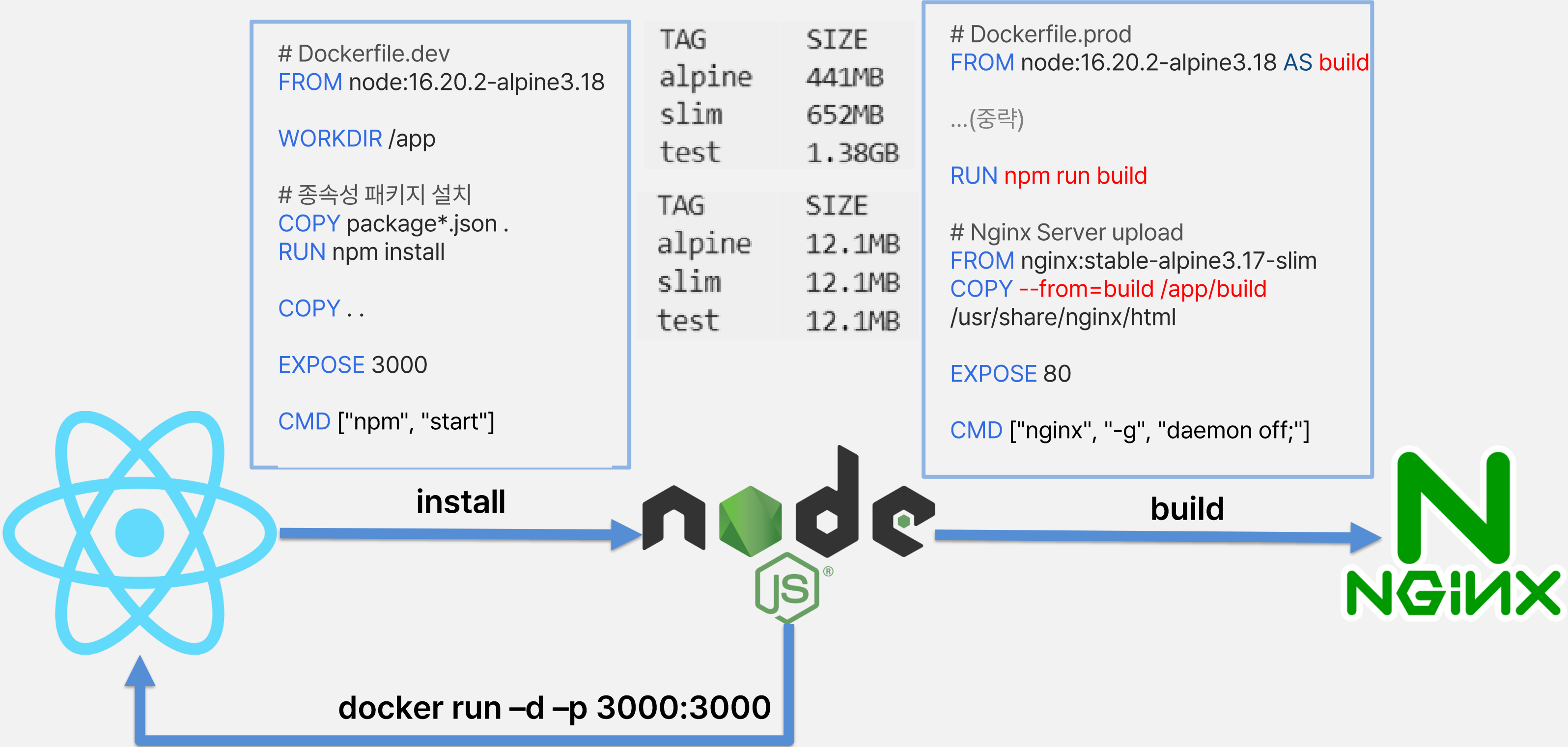
dev

- ✓ 개발한 애플리케이션을 **빠르게 컨테이너화**
- ✓ 이미지 **경량화**

prod

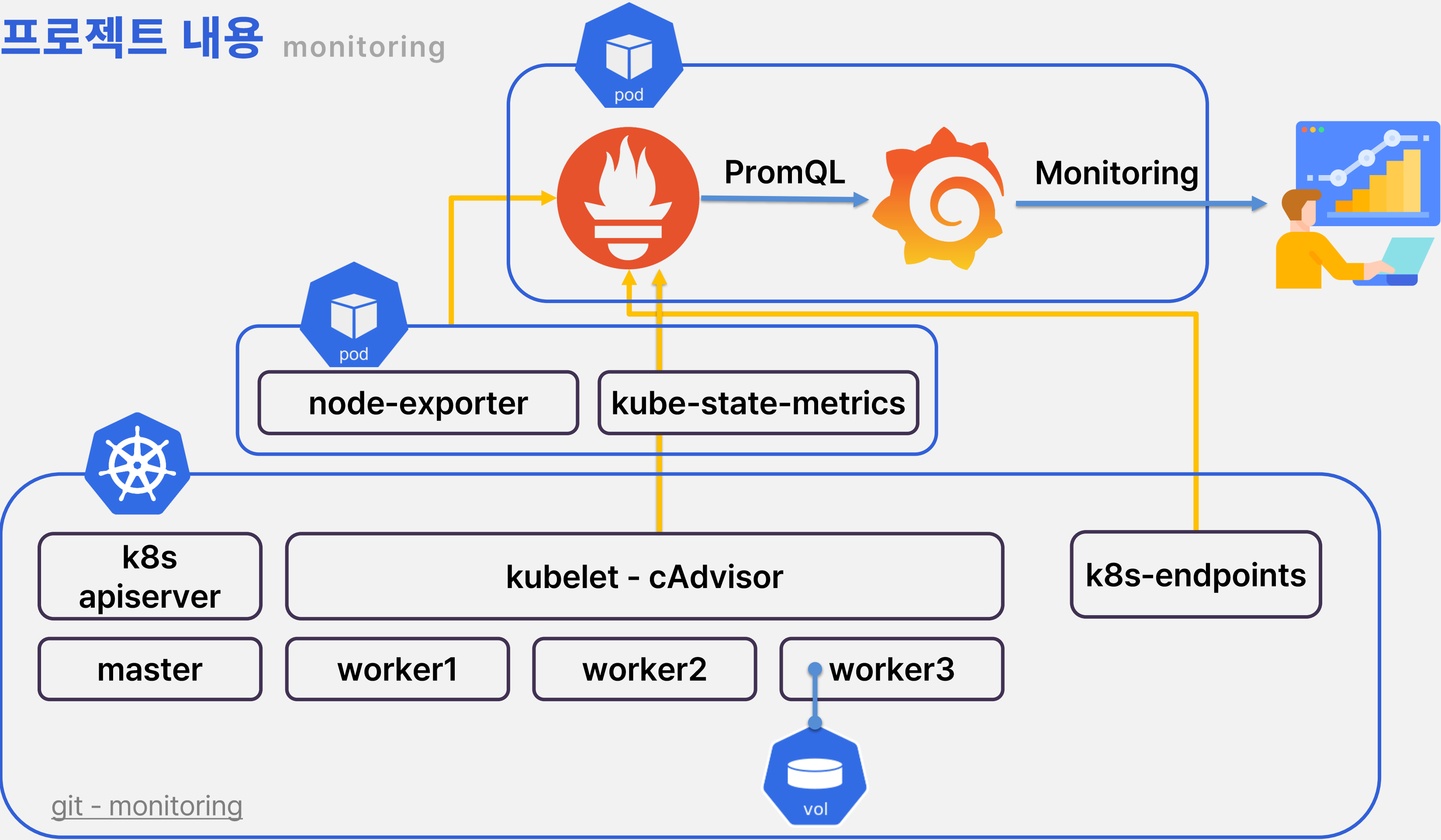
- ✓ 반복적인 CI 테스트를 거친 이후 **배포단계**에 쓰기 위한 컨테이너 이미지 생성
- ✓ 애플리케이션의 **빌드 과정** 추가







측면	첫 번째 접근 방식 (그라파나 및 프로메테우스를 마스터 및 일부 워커 노드에 설치)	두 번째 접근 방식 (그라파나 및 프로메테우스를 전용 모니터링 노드에 설치)
클러스터 구조	클러스터에 분산	클러스터와 분리
자원 사용 및 성능	자원 사용, 성능에 영향 0	해당 노드에 자원을 할당, 다른 노드에는 영향 X
유지보수 및 관리	유지보수 작업이 다른 작업에 영향	유지보수 작업이 해당 노드에만 영향
보안	다른 작업과 공유되어 있어 보안에 위협	전용 노드에 설치되어 보안 강화



수행 과정

- ✓ 프로메테우스와 그라파나를 **클러스터에 배포**하고 연동하는 작업 수행
클러스터의 상태와 지표를 수집하고, 그라파나를 사용하여 **시각화**하는 대시보드 구축
- ✓ 쿠버네티스에서 제공하는 오토스케일링 기능을 활용하여 파드의 **동적 확장** 구성
파드의 스케일 인/아웃 이벤트를 감지, 이를 그라파나 대시보드에 **실시간 반영**
- ✓ 프로메테우스가 수집한 데이터를 그라파나 대시보드에 시각적으로 표현하는 작업 수행
오토스케일링 이벤트와 관련된 데이터를 **수치화**하여 파드의 동적 확장 상황 **실시간 확인**

기능 구현

- ✓ 프로메테우스가 수집한 데이터 기반으로 자동 대시보드 업데이트 기능 구현
새로운 지표나 이벤트가 발생할 때마다 대시보드는 **실시간으로 갱신, 최신 정보를 제공**
 - ✓ 클러스터 및 애플리케이션에서 발생하는 이벤트를 지속적으로 로깅, 분석하는 기능 구현
이를 통해 시스템의 **동작에 대한 로그 수집**, 문제 발생 시 원인을 파악 후 대응
-

사진2)

오토스케일링을 통해 파드의 동적 확장을 실시간으로 감지하고, 그 수치화된 결과를 시각화하여 대시보드에 투영
이로써, 인프라의 투명한 시각화와 명확한 데이터 분석을
통해 시스템 확장의 효율성과 신속성을 확인

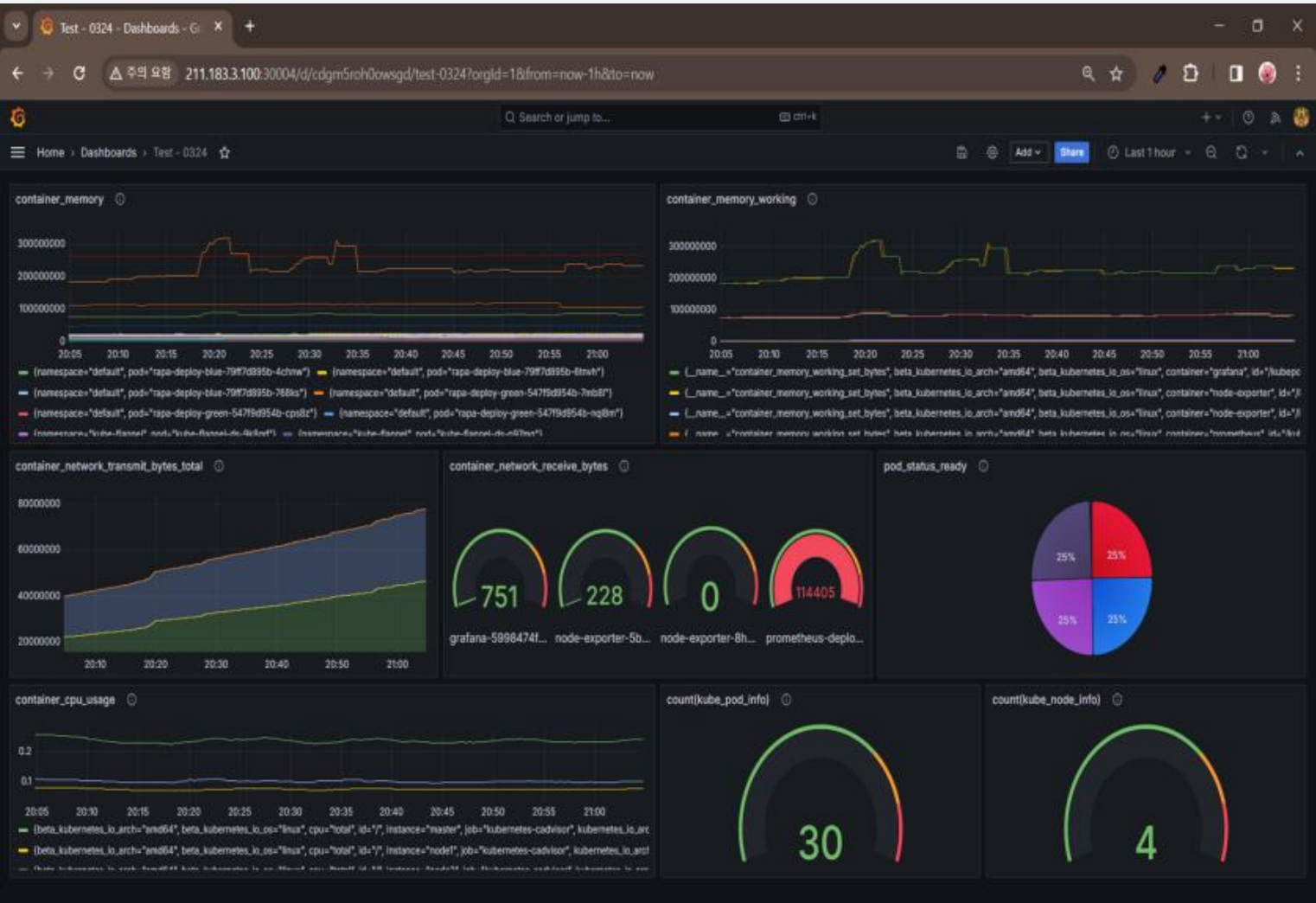


사진1)

사진1)

파드의 동적 확장에 대한 실행 전 상태
설정하기 전의 상태를 파악하여 이후의 변경을 추적하고,
시스템의 성능 및 확장에 대한 변화 적용



사진2)



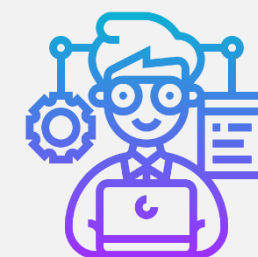
윤순상(PM)

젠킨스 & 소나 큐브,
CI 파이프라인



유성욱

k8s, ArgoCD,
CD 파이프라인



김태교

웹 (React)
Dockerfile



유태균

모니터링
프로메테우스 & 그라파나



권재성

k8s, 모니터링
프로메테우스 & 그라파나

카무카무's 간트차트

2024/ 03.15 ~ 04.04

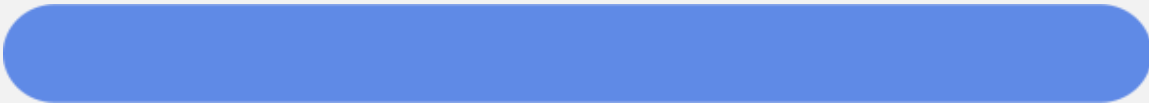
all

김태교

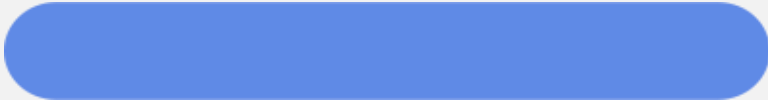
윤순상
유성욱

권재성
유태균

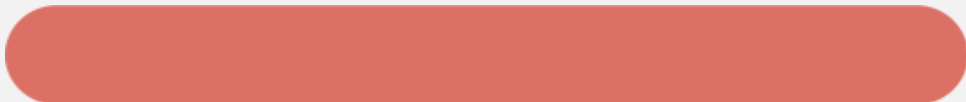
Project
Plannig



Infra Setup



Application



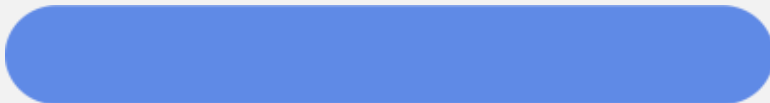
CI/CD



Monitoring



PPT



15 18 19 20 21 22 25 26 27 28 29 1 2 3 4



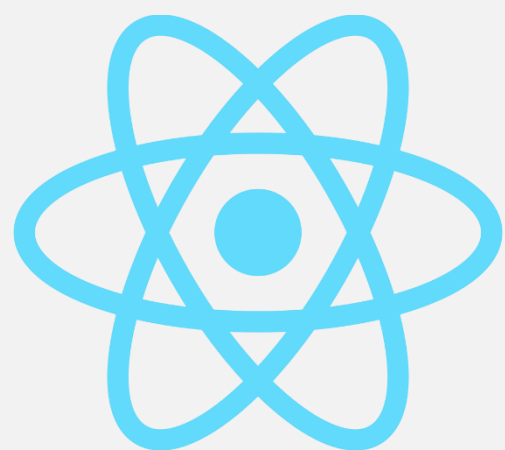
적극적인 중간회의

주기적인 중간회의를 통해 방향성을 잃지 않고 서로의 과정을 피드백하며 협업할 수 있었다.



소극적인 협업툴 사용

기획했던 협업툴을 이용한 회고록 작성, 일정 조율, 작업물 체크 등 협업툴의 이점을 잘 활용하지 못했다.



Application

성능 향상

- ✓ 코드 스플리팅, 렌더링 최적화를 통해 성능을 더욱 향상

컨텐츠

- ✓ 다양한 컨텐츠를 추가하여 다채로운 블로그를 구성



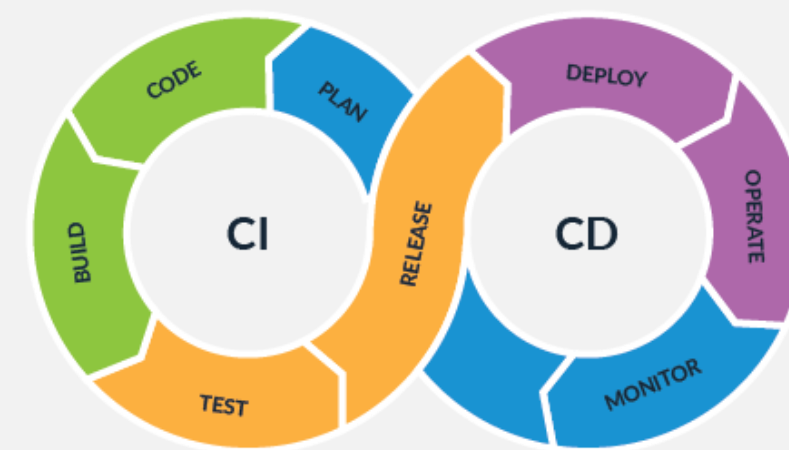
Monitoring

내구성 & 가용성

- ✓ 데이터의 가용성을 위해 백업 옵션을 제공

관리 용이성

- ✓ 데이터 보안을 강화하기 위한 다양한 기능을 제공하여 관리



CI & CD

Jenkins Pipeline

- ✓ 특정 파일의 커밋 변경만 감지

Jenkins Pod

- ✓ k8s 환경에 알맞은 CI 자동화
- ✓ kustomize 활용

감사합니다.

Go Project Github

2024.04.04

RAPA KAKAO
Engineer 4