



# 당근 개발자 플랫폼은 어떤 문제를 해결하고 있는가?

2024.1.10 @ AWSKRUG 플랫폼엔지니어링모임  
Outsider



# Outsider

## 당근마켓 SRE

[Outsider's Dev Story](#) 블로그

[44BITS 팟캐스트](#)

[RetroTech 팟캐스트](#)



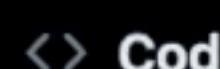
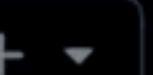
# Kubernetes

**Platform Engineering이  
목표는 아니다**



daangn / wayland-demo

Type / to search



Issues

Pull requests

Actions

Projects

Wiki

Security

5

Insights

Settings

wayland-demo

Internal

Edit Pins

Unwatch 1

Fork 0

Star 0

main

1 branch

0 tags

Go to file

Add file

Code

## About

No description, website, or topics provided.

Activity

0 stars

1 watching

0 forks

## Releases

No releases published

Create a new release

## Packages

No packages published

Publish your first package

outsideris initial commit ...

509dc7a 5 hours ago 1 commit

bin	initial commit	5 hours ago
public/stylesheets	initial commit	5 hours ago
routes	initial commit	5 hours ago
views	initial commit	5 hours ago
.gitignore	initial commit	5 hours ago
Dockerfile	initial commit	5 hours ago
app.js	initial commit	5 hours ago
package-lock.json	initial commit	5 hours ago
package.json	initial commit	5 hours ago

Add a README with an overview of your project.

Add a README

## Languages



# wayland-demo

[Deployments](#) [Workloads](#) [CronJobs](#) [Configs](#) [Resources](#) [Cost](#) [Settings](#)

## Deployments

[?](#) [배포](#) [Rollback ▾](#)

### 배포 기록이 없어요

배포를 진행하려면 다음 과정을 진행해 주세요.

### 배포 Manifest

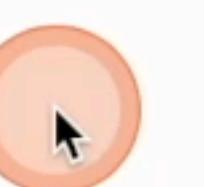
배포 구성을 설정하는 배포 Manifest 파일 `.deploy/values.yaml`을 저장소에 생성해야 합니다.

배포 Manifest는 아래와 같은 형식이며 자세한 내용은 [배포 미리보기](#)에서 볼 수 있습니다.

아래 예시를 참고하여 프로젝트에 맞게 수정해서 사용하세요.

[GitHub에서 바로 작업하기](#)

```
# yaml-language-server: $schema=https://  
  
type: v1/appservice  
images:  
  app:  
    dockerfile: Dockerfile  
  
services:  
  app:  
    image: app  
    class: k1.small  
    ports:  
      - name: http
```



### About

프로젝트 설명이 없습니다.

[등록하기](#)

 [등록하기](#)

### Team

담당 팀: 딜리버리 파트

슬랙 그룹: sre-delivery

슬랙 채널: \_infra-sre

팀 문서: [딜리버리 파트 문서](#)



2021년

**SRE팀 딜리버리 파트**

저는 2020년 12월 입사

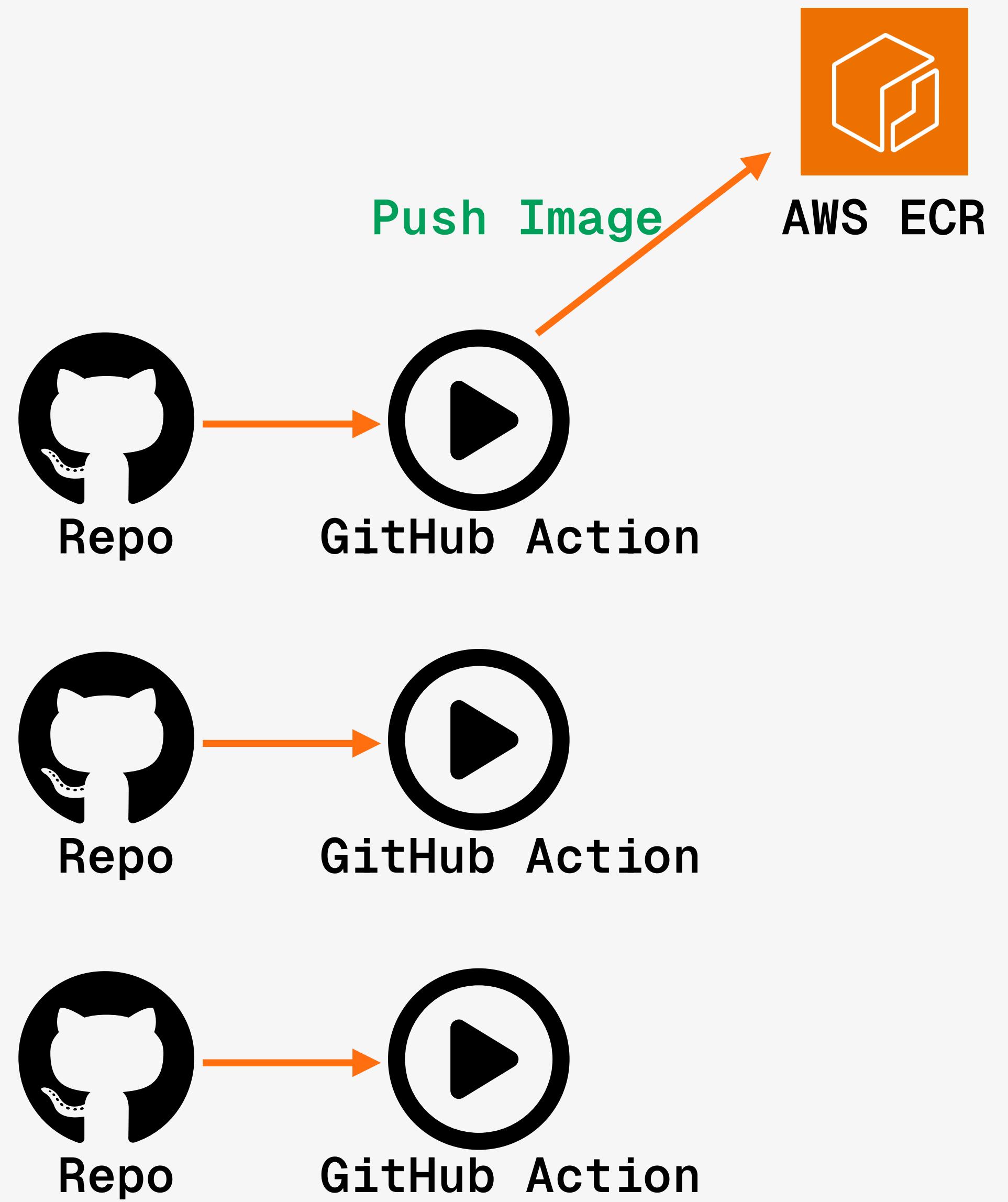


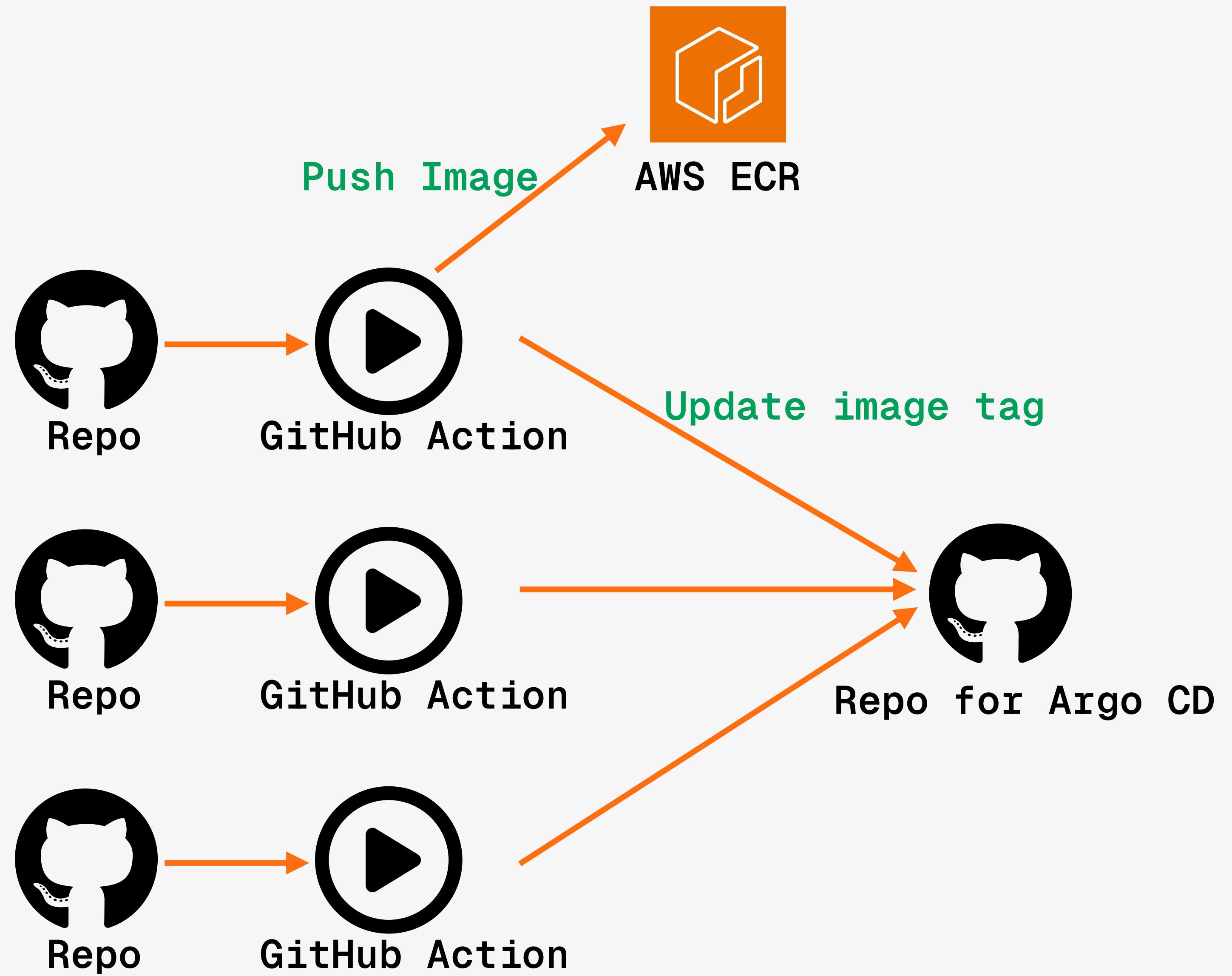
**Argo CD**

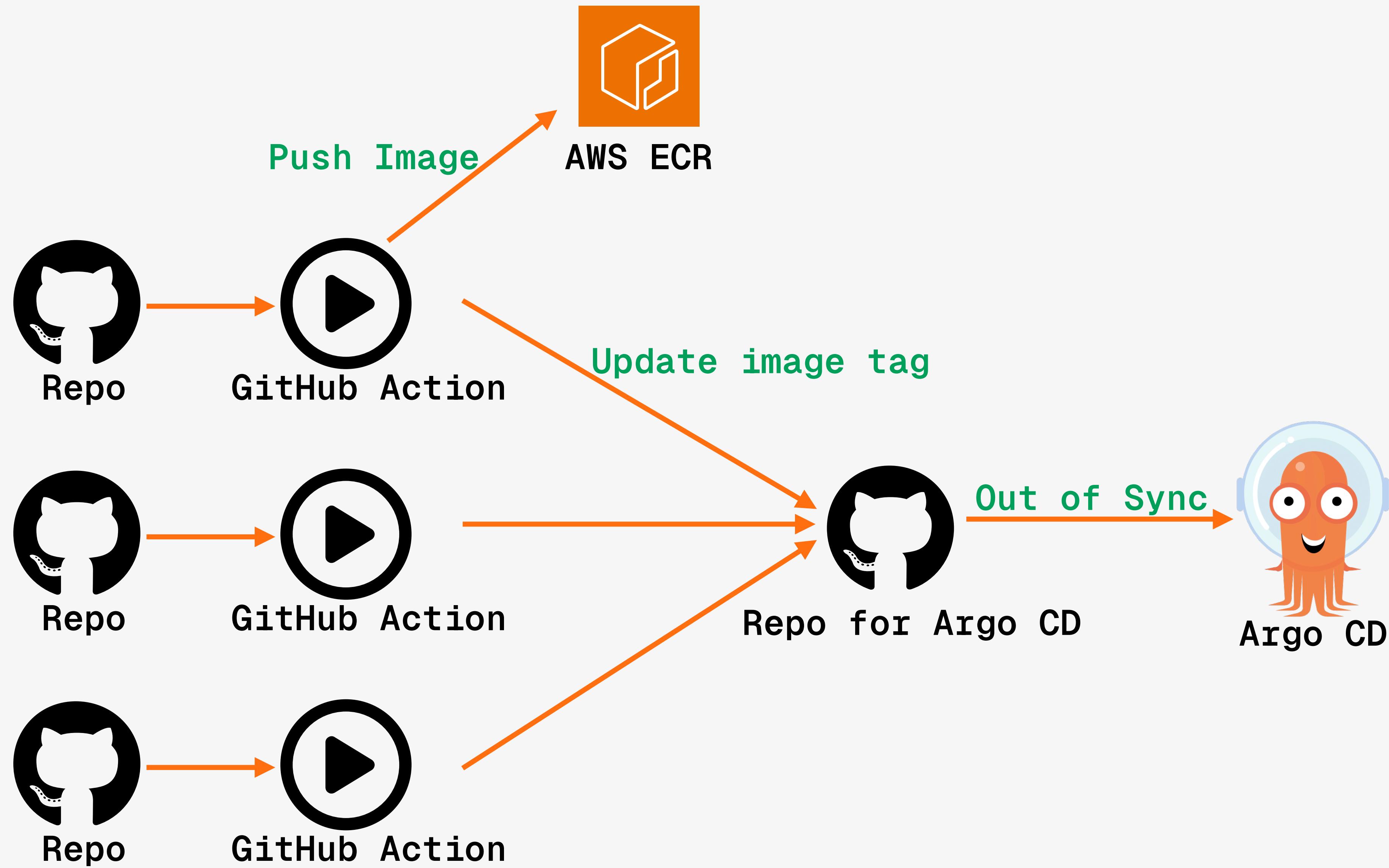
+

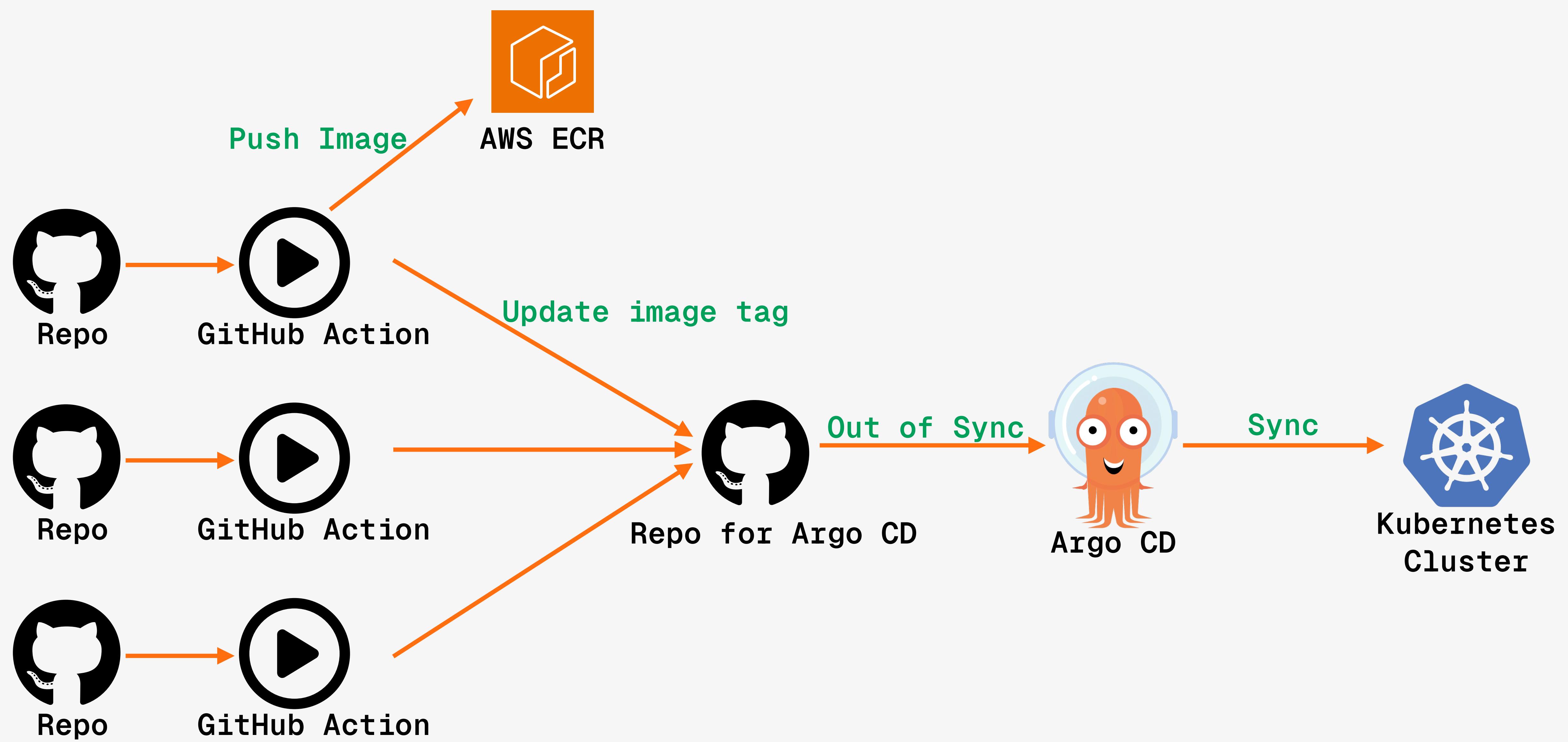
**Kustomize**

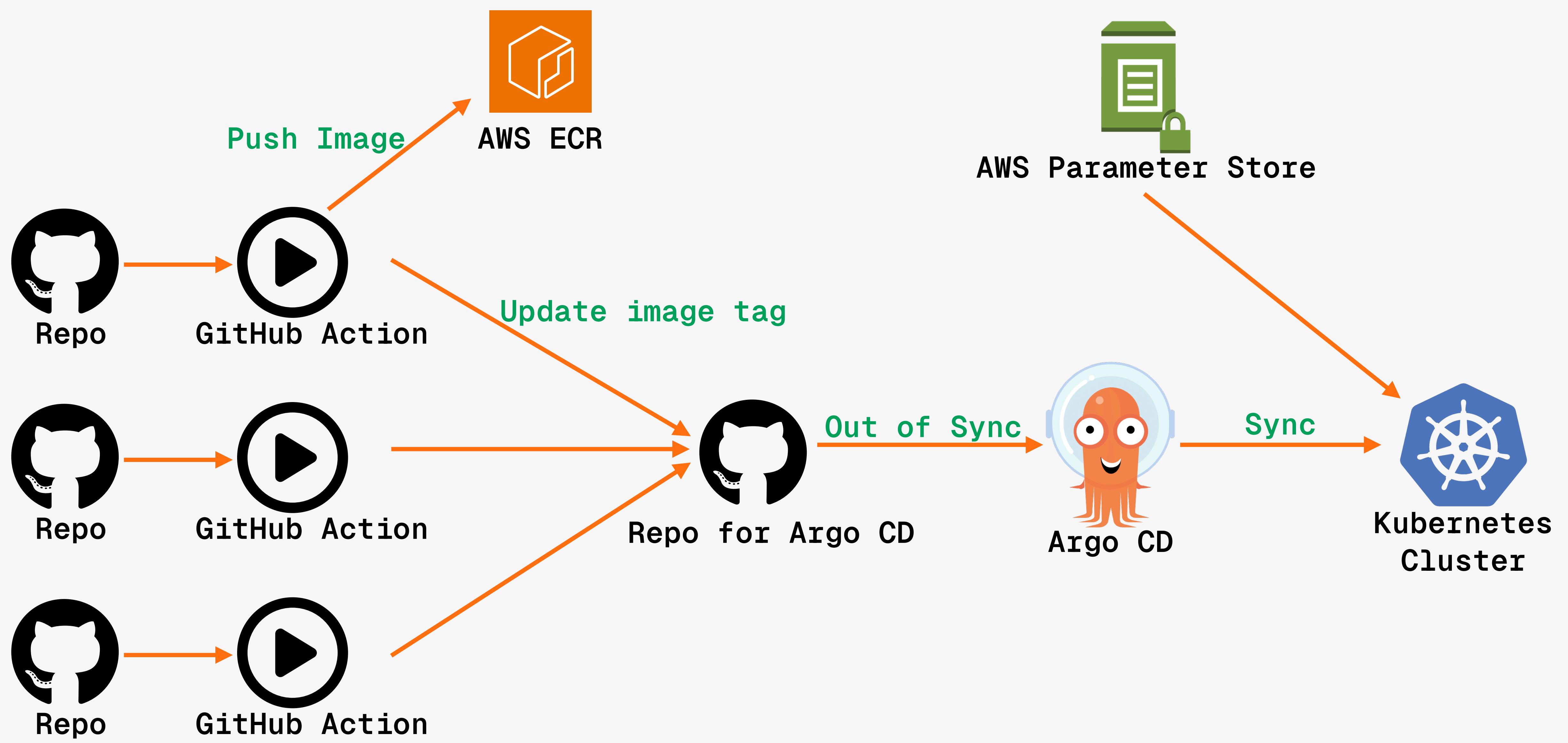


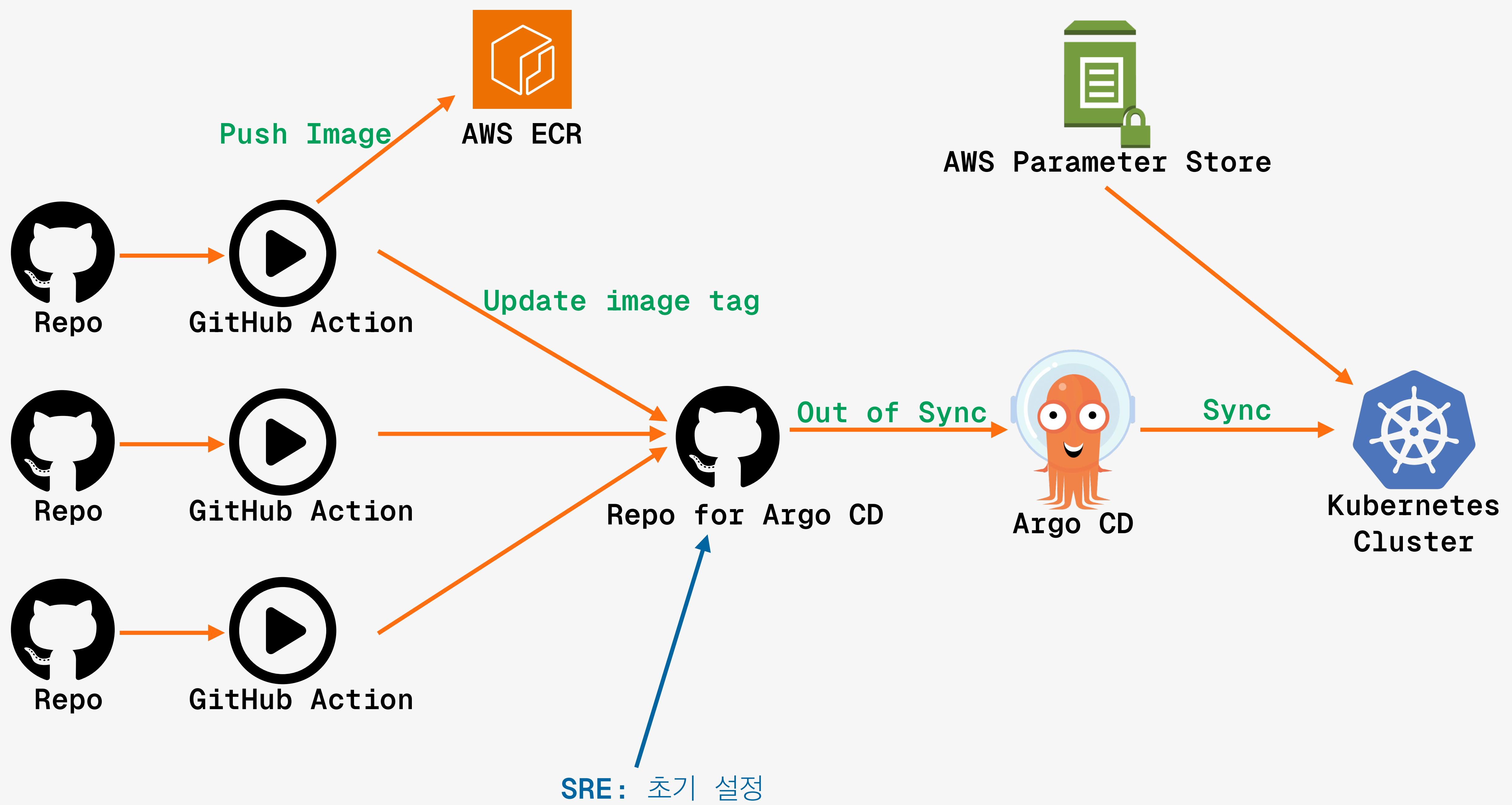


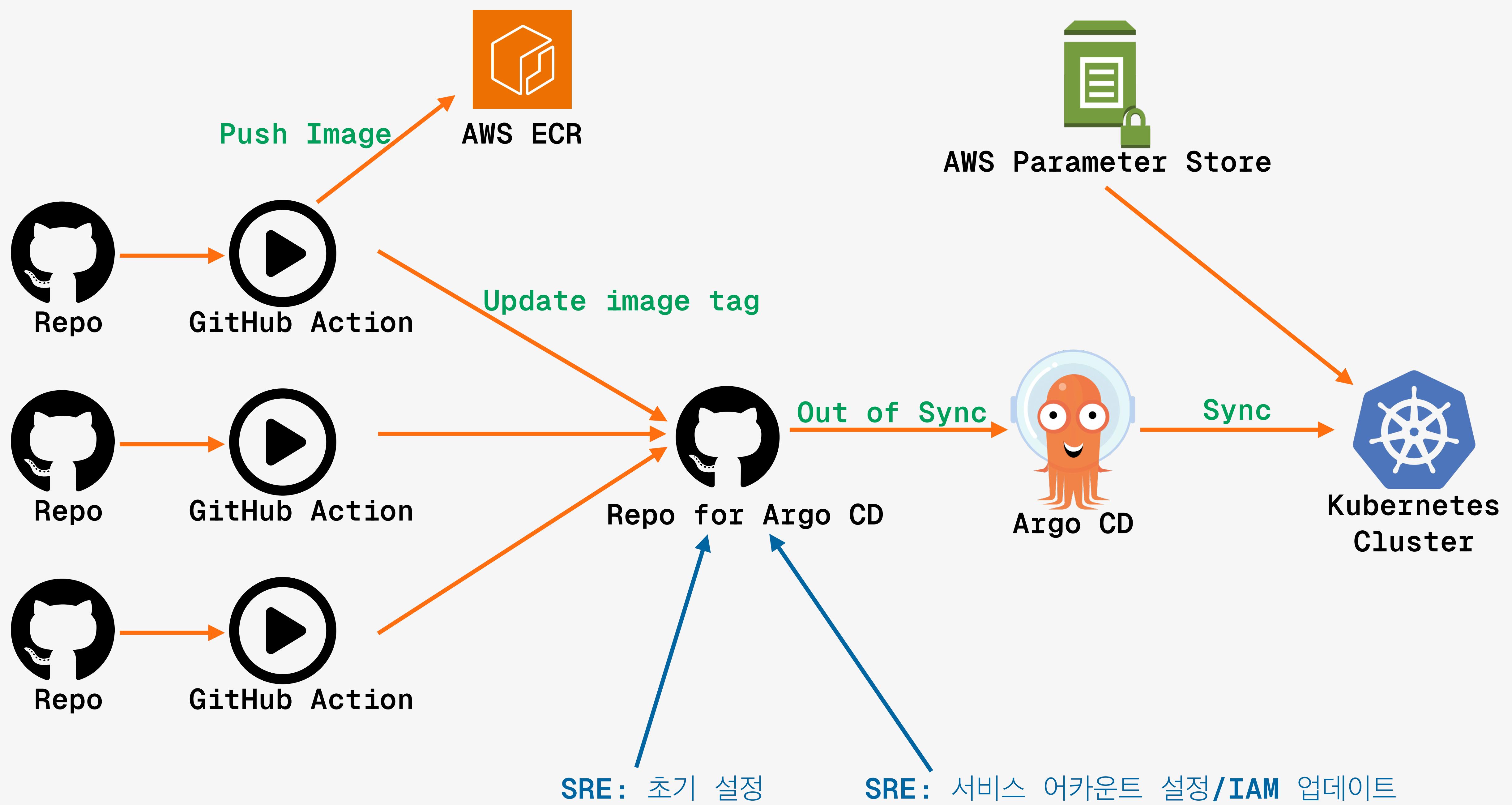














# Fragmentation 파편화

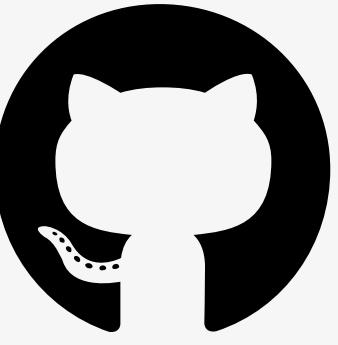
사진: [Unsplash](#)의 [Kelly Sikkema](#)



# Fragmentation 파편화

50+

사내  
엔지니어



Repo for Argo CD



SRE



# Separation of Concerns

사진: [Unsplash의 Alex Padurariu](#)

**수정하기 어렵다**

# **반복되는 배포 요청**



사진: [Unsplash](#)의 Alexander Simonsen

고쳐쓸까?

새로 만들까?



**Argo CD를 버린다**



**좋은 배포란 무엇인가?**

누가

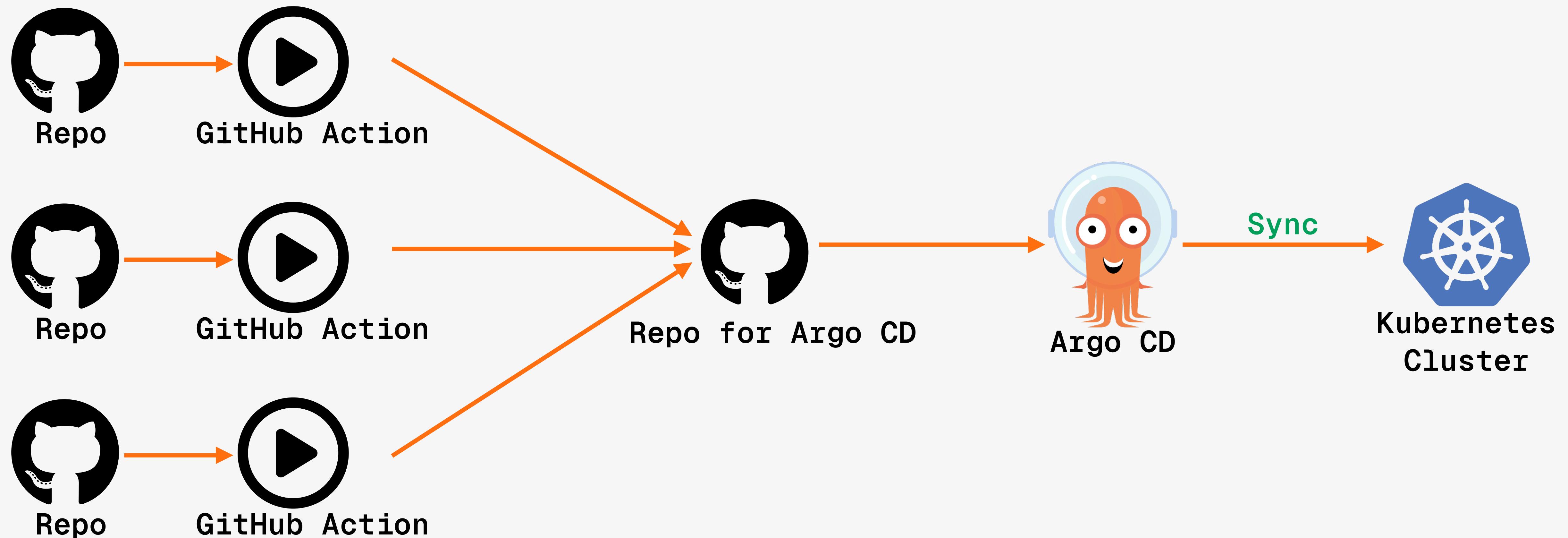
---

언제

---

무엇을 배포했는가?

---



 Deployed At:

2 days ago (Fri Jan 05 2024  
18:25:02 GMT+0900)

 Time to deploy:

01:28 min

Active for:

2 days 01:55 hours

Revision:

Authored by

[539bf38](#)

github-merge-queue[bot] <null>

2 days ago (Fri Jan 05 2024 18:23:21 GMT+0900)

ci(prod/kr/kontrol): update version 0476d06c0d9323a6631feaf2360e

GPG signature

-

KUSTOMIZE

VERSION

default

IMAGES

docker.io/demo

docker.io/demo

curlimages/curl

curlimages/curl

 Deployed At:

2 days ago (Fri Jan 05 2024  
17:20:53 GMT+0900)

 Time to deploy:

01:29 min

Revision:

[b9d1cb5](#)

⋮

⋮

## Commit

**ci(prod/kr/kontrol): update version 0476d06c0d9323a6631feaf2360e9649a...**

[Browse files](#)

...7109388

feat/mlrecs + release/prod

github-merge-queue[bot] committed 2 days ago

1 parent afd03c2 commit 539bf38

Showing 1 changed file with 1 addition and 1 deletion.

[Whitespace](#)

[Ignore whitespace](#)

[Split](#)

[Unified](#)

# Commit

✓ cdk8s: update version v1.2.89 (#1853)

[Browse files](#)

Co-authored-by: b6pzeusb

↳ main (#1853)

🏷️ 2024-01.v7

 daangnbot and b6pzeusb

committed 2 days ago

Verified

1 parent 0dc4486

commit 0476d06

Showing 2 changed files with 5 additions and 5 deletions.

[Whitespace](#)

[Ignore whitespace](#)

[Split](#)

[Unified](#)

누가  
어제  
누구든 쉽게 알 수 있어야  
누구든 쉽게 알 수 있어야  
매포했는가?



**상태 동기화**

**desired state = actual state**

**최종 일관성**

**Eventually Consistency**

# v1 운영 중



v1 운영 중

v1

v2로 변경 제출

## **desired state 동기화**

**v1 운영 중**



**v1**



**v2로 변경 제출**



**desired state 동기화**

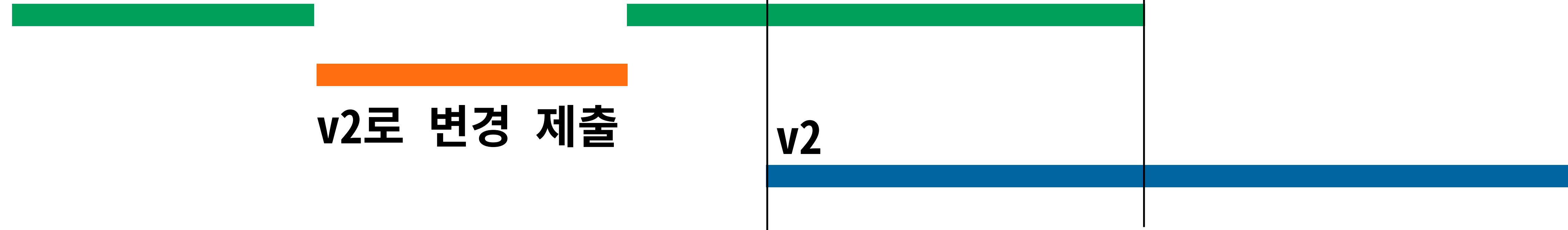
**v1 운영 중**

**v1**

**v2로 변경 제출**

**v2**

**rollout 완료**



**desired state 동기화**

**v1 운영 중**

**v1**

**v2로 변경 제출**

**v2**

**rollout 완료**

**Argo CD의 동기화 완료**

**desired state 동기화**

**v1 운영 중**

**v1**

**v2로 변경 제출**

**v2**

**rollout 완료**

**Argo CD의 배포 완료**

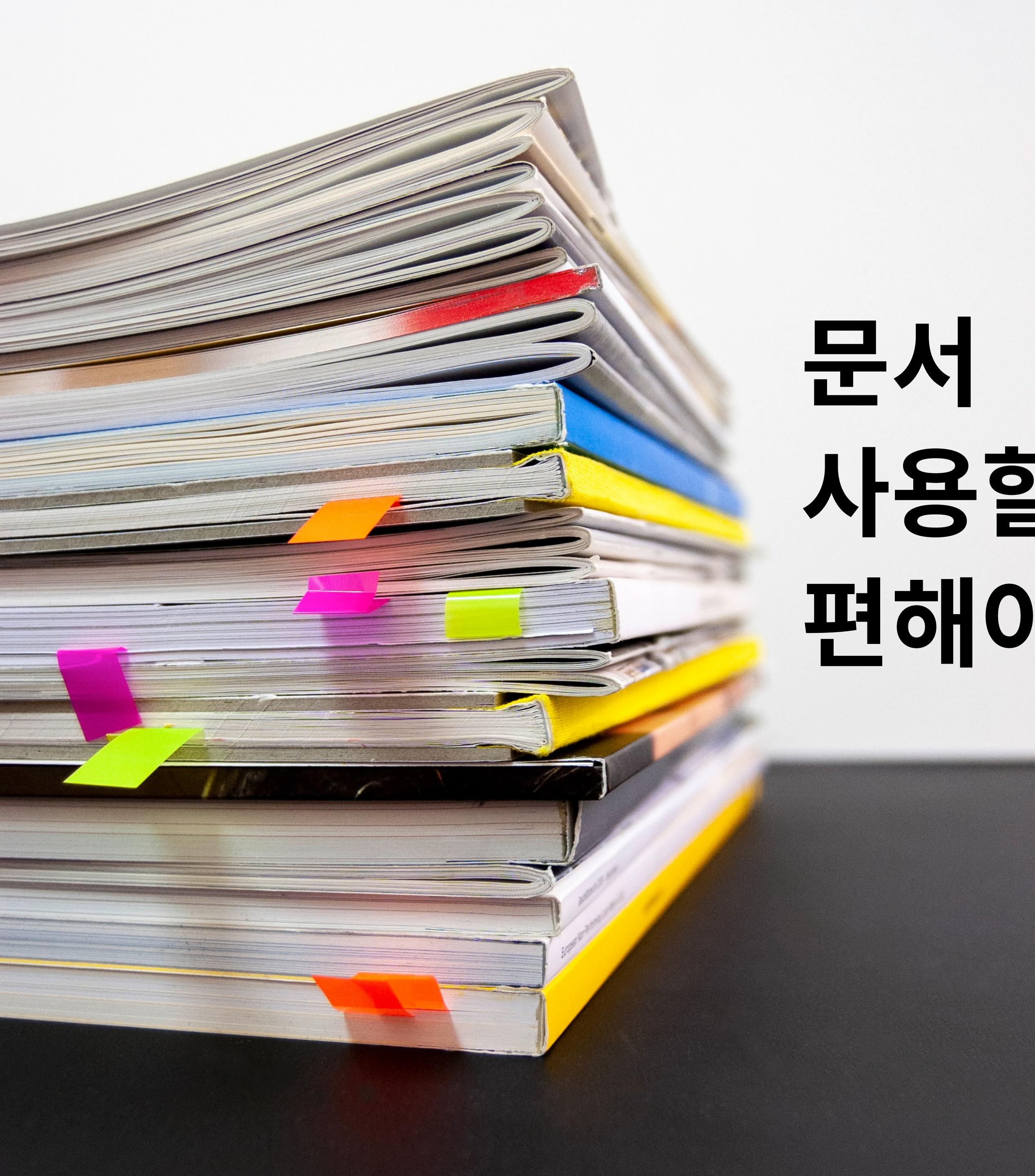
**엔지니어가 기대하는 배포 완료**

# Kubernetes를 몰라도 배포할 수 있어야 한다

Kubernetes를 알면 더 잘 쓸수 있다

**오늘 입사한 사람도 배포할 수 있어야 한다**

Self-Service  
**처음부터 끝까지 직접 할 수 있어야 한다**



**문서 없이도  
사용할 수 있을 정도로  
편해야 한다**

celpax

How was  
your day?



Together we can improve our workplace

Your logo here

설문  
조사는  
하지  
않습니다

**사람들은  
원인이 아니라  
해魑책을 가져온다**



**Opinionated**

**단순함**



**유연함**

**Convention  
over  
Configuration**

90%

**Kubernetes manifest 추상화**

# **Configuration files**

# Cloud Development Kit for Kubernetes

cdk8s is an open-source software development framework for defining Kubernetes applications and reusable abstractions using familiar programming languages and rich object-oriented APIs. cdk8s apps synthesize into standard Kubernetes manifests which can be applied to any Kubernetes cluster.

The screenshot illustrates the synthesis process between two code editors. On the left, the TypeScript code for a chart is shown:

```
cdk8s-welcome > TS main.ts > MyChart > constructor > image
1 import { Construct } from 'constructs';
2 import { App, Chart } from 'cdk8s';
3 import { WebService } from './lib/web-service';
4
5 class MyChart extends Chart {
6   constructor(scope: Construct, name: string) {
7     super(scope, name);
8     new WebService(this, 'aws', {
9       options: WebServiceOptions,
10      image: 'aws/webapp:1.6'
11    })
12  }
13}
14
15 const app = new App();
16 new MyChart(app, 'cdk8s-welcome');
17 app.synth();
18
```

On the right, the resulting Kubernetes manifest is displayed:

```
cdk8s-welcome > dist > ! cdk8swelcome.k8s.yaml > {} spec > {} template > {} spec > [ ] co
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: cdk8s-welcome-aws-webapp-service-2325fc6b
5 spec:
6   ports:
7     - port: 80
      targetPort: 8080
      type:
        app: cdk8swelcomeawswebapp83EC99A6
      type: LoadBalancer
12
13 apiVersion: apps/v1
14 kind: Deployment
15 metadata:
16   name: cdk8s-welcome-aws-webapp-deployment-92243c39
17 spec:
18   replicas: 1
19   selector:
20     matchLabels:
21       app: cdk8swelcomeawswebapp83EC99A6
22   template:
23     metadata:
24       labels:
25         app: cdk8swelcomeawswebapp83EC99A6
```

**어디서부터 배포라고 할 것인가?**

코딩

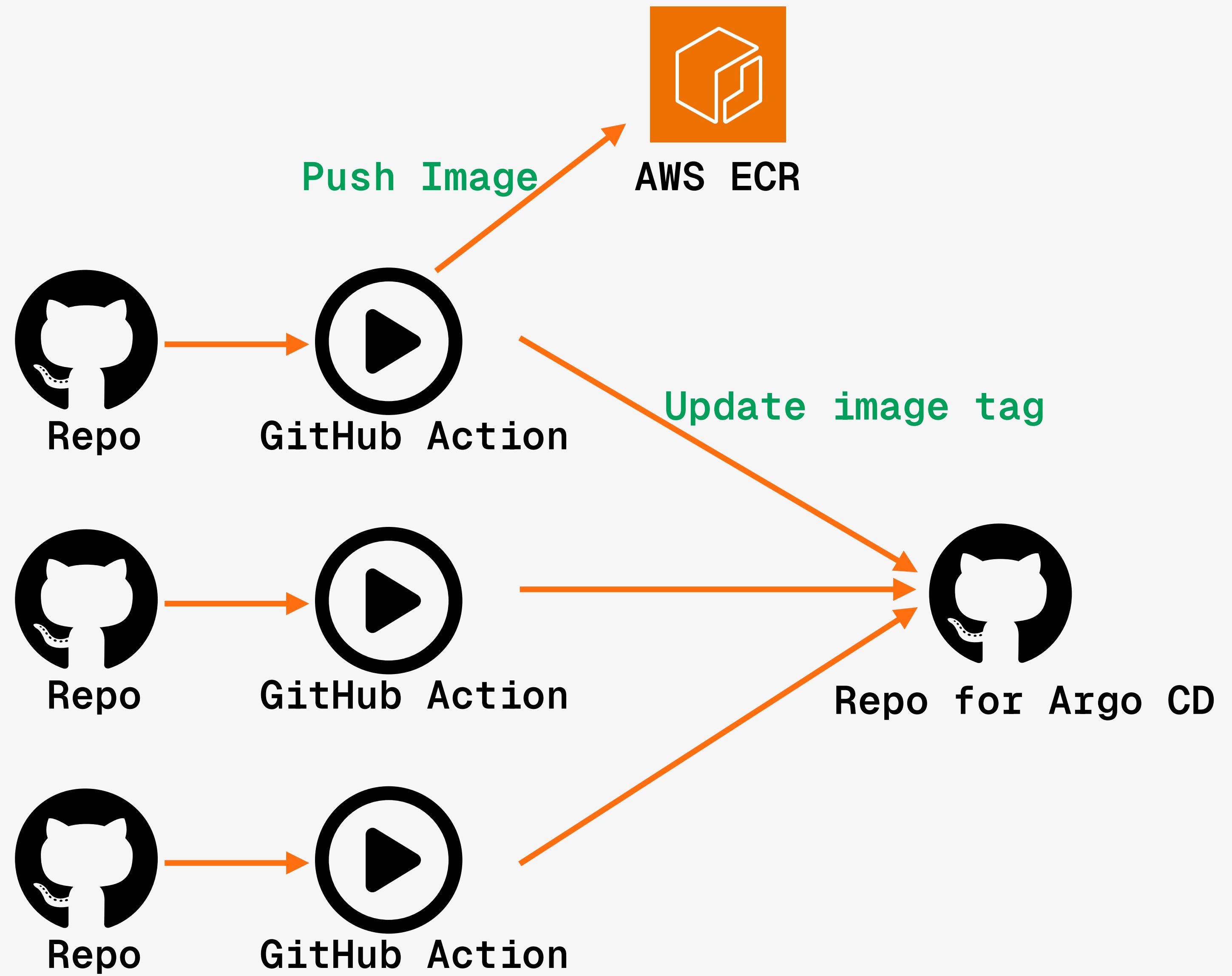
테스트

Docker  
빌드

Docker  
푸시

Kubernetes  
manifest

kubectl  
apply



코딩

테스트

Docker  
빌드

Docker  
푸시

Kubernetes  
manifest

kubectl  
apply

**GitHub**

**Argo CD**

코딩

테스트

Docker  
빌드

Docker  
푸시

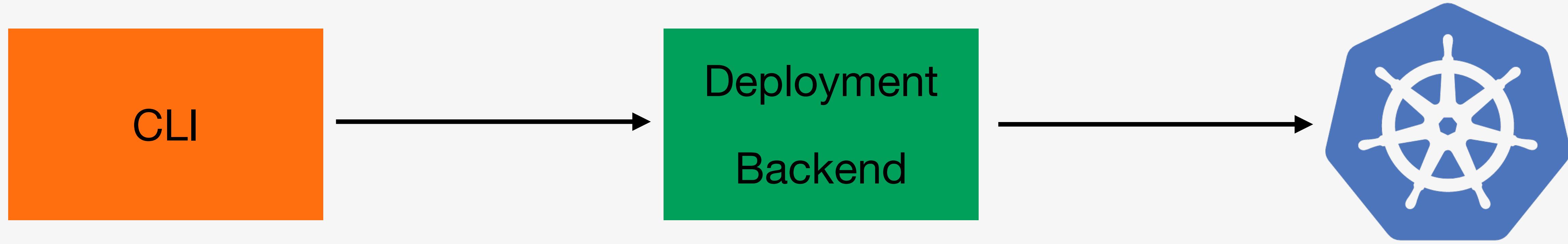
Kubernetes  
manifest

kubectl  
apply

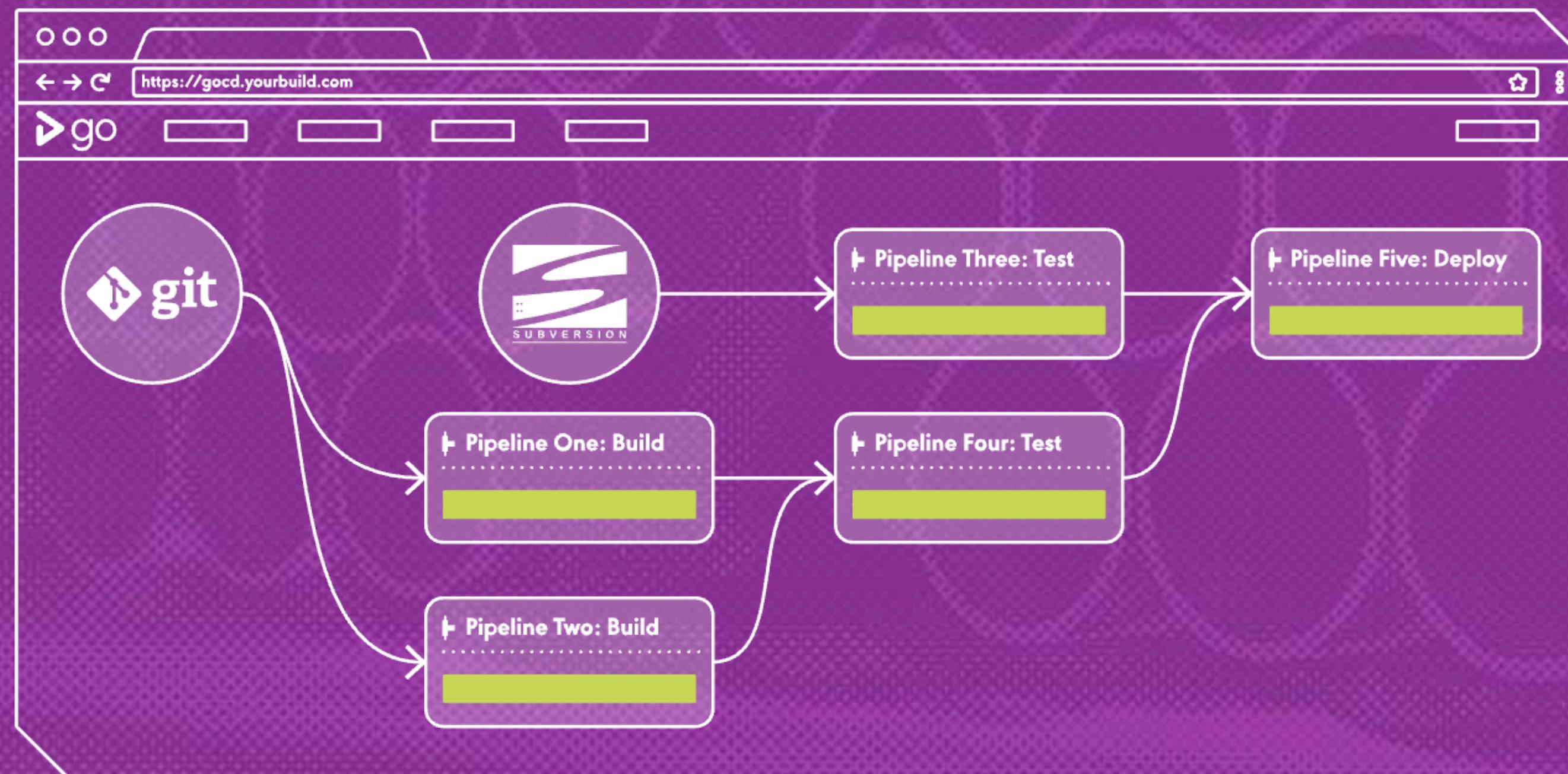


**GitHub**

**새 배포 시스템**

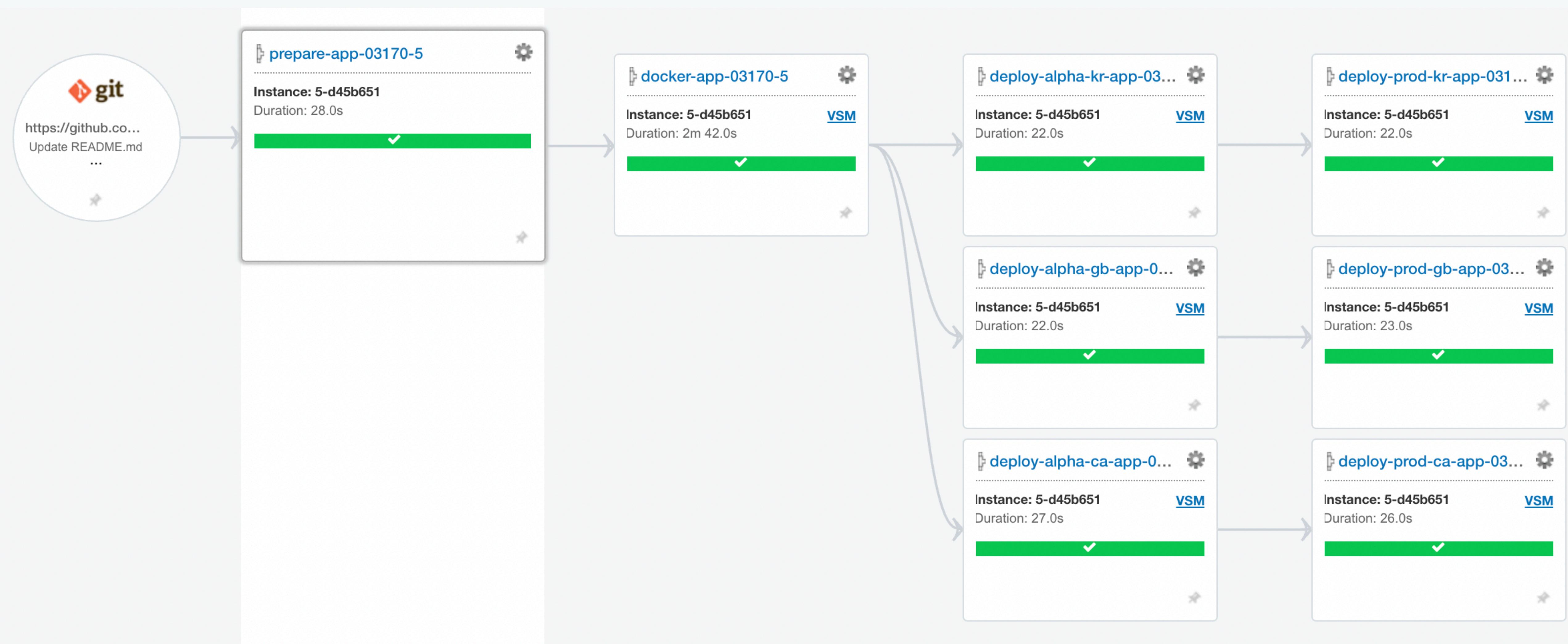


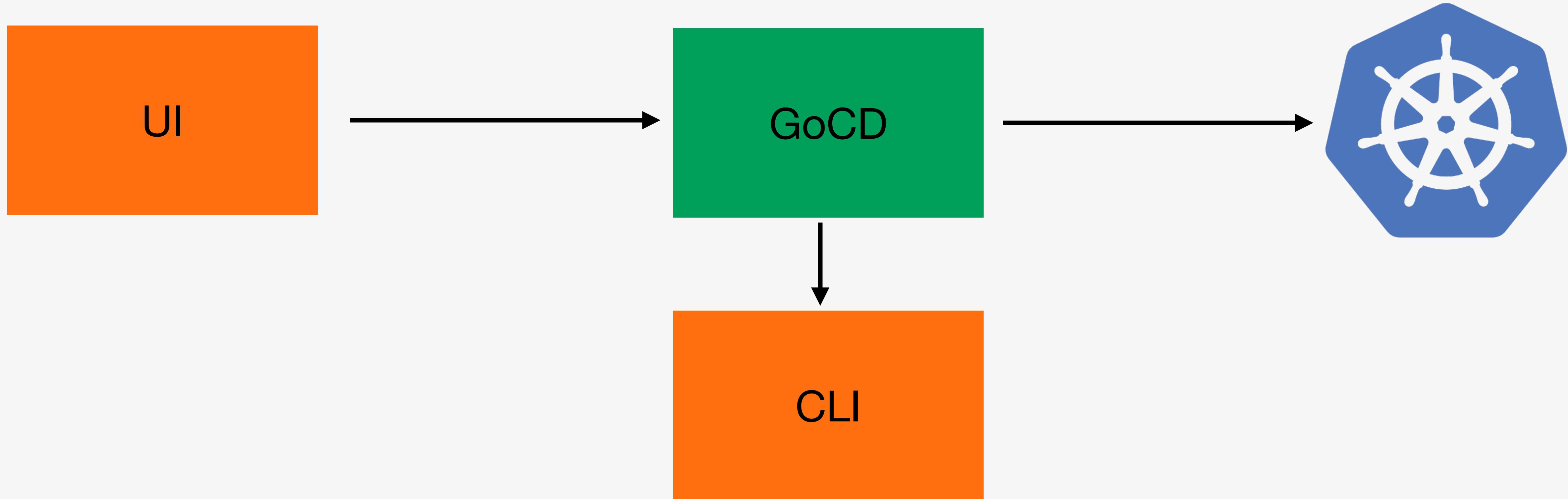
# FREE & OPEN SOURCE CI/CD SERVER



Easily model and visualize complex workflows with GoCD.

[TEST DRIVE GOCD](#)





# Mockup



## 프로젝트 등록

프로젝트 app-0318-1

GitHub 주소 <https://github.com/outside-the-box-kr/app-0318-1>

Submit

- o 파이프라인 그룹 생성
- o prepare 파이프라인 생성
- o docker 파이프라인 생성
- o alpha-kr 파이프라인 생성
- o prod-kr 파이프라인 생성
- o 파이프라인 생성 완료



# app-0318-1

배포

## Pipeline

리전 추가

prepare

docker

deploy-alpha-kr

deploy-prod-kr

## History

### Build 2-d45b651

상세 보기

Triggered by [@wayland](#) on 2021. 3. 18. 오후 6:59:41 with [d45b651](#)

● prepare

● docker

● deploy-alpha-kr

● deploy-prod-kr

### Build 1-d45b651

상세 보기



# app-0318-1

## Pipeline

prepare

배포

X

배포

특정 커밋으로 배포하려면 Git의 Sha를 입력하세요. 입력하지 않으면 최신 커밋으로 배포합니다.

배포할 Git 커밋

SHA

배포하기

리전 추가

deploy-prod-kr

## History

### Build 2-d45b651

Triggered by [@wayland](#) on 2021. 3. 18. 오후 6:59:41 with d45b651

상세 보기

# Deployment 2-d45b651

[Open](#)

Triggered with d45b651



Wayland triggered prepare pipeline on 2021. 3. 18. 오후 6:59:41 for 31 secs

[로그](#)

Wayland triggered docker pipeline on 2021. 3. 18. 오후 7:00:21 for 169 secs

[로그](#)

Wayland triggered deploy-alpha-kr pipeline on 2021. 3. 18. 오후 7:03:21 for 23 secs

[로그](#)[manifest](#)

```
##|10:03:23.121 [go] Received request to create a pod for job JobIdentifier{pipelineName='deploy-alpha-kr', deploymentName='app-0318-1', deploymentId='2-d45b651', stage='alpha', target='kr'}
```

```
##|10:03:23.128 [go] Waiting to create agent pod.
```

```
##|10:03:23.163 [go] Creating pod: gocd-agent-25751021-75bb-49f6-83c9-3e243898e5a9
```

```
##|10:03:23.163 [go] Agent pod gocd-agent-25751021-75bb-49f6-83c9-3e243898e5a9 created. Waiting for pod to start.
```

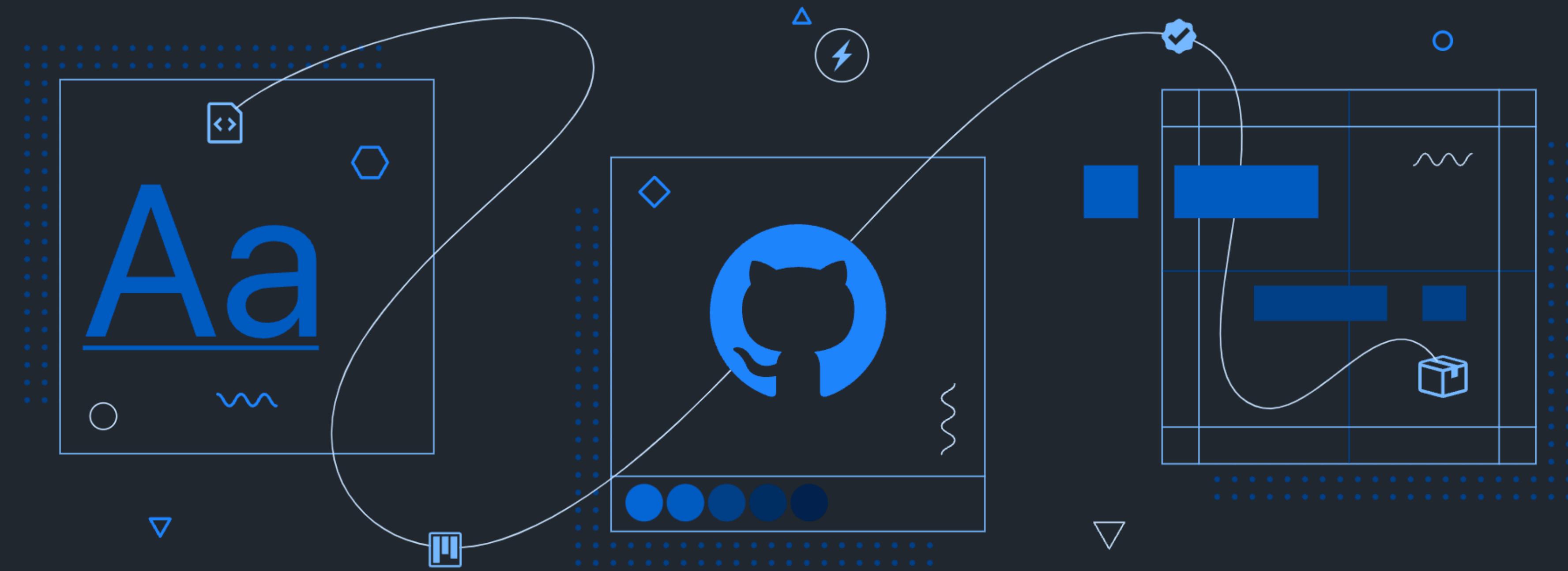
```
##|10:03:43.561 [go] Job Started: 2021-03-18 10:03:43 GMT
```

```
##|10:03:43.561
```

```
pr|10:03:43.562 [go] Start to prepare deploy-alpha-kr-app-0318-1/2-d45b651/deploy/1/k8s on gocd-agent-25751021-75bb-49f6-83c9-3e243898e5a9
```

```
pr|10:03:43.571 [go] Start to update materials.
```

# Primer Design System



Primer is a set of guidelines, principles, and patterns for designing and building UI at GitHub. It provides a shared language and standardized approach to delivering cohesive experiences.

## Guides

Standards, guidelines, and tools to getting started with

## Foundations

The fundamental parts of the design system that underpin

<https://primer.style/>

# Improving how we deploy GitHub

As GitHub doubled its developer head count, tooling that worked for us no longer functioned in the same capacity. We aimed to improve the deployment process for all developers at GitHub and mitigate risk associated with deploying one of the largest developer platforms in the world.

The screenshot shows a GitHub pipeline status page for a 'production\_rollout' pipeline. The top navigation bar includes 'apps / github / pipelines / production\_rollout / #945'. The main title is 'production\_rollout #945'. A green button indicates 'Completed Success' with the message 'jules2689 deployed train-abcd123 to production'. To the right is a user profile picture and a 'Lock' button. The pipeline timeline shows three steps: 'Step #1: Preparing deployment' (green checkmark), 'Step #2: Deploy to canary' (green checkmark, subtext: 'Completed in 4m 57s • Started 3 months ago'), and 'Timer Gate (5m)' (green checkmark). On the right side, there's a sidebar with links to 'Splunk Logs', '#dotcom-ops', and 'Debug'. Below that is a section titled 'Extra Context' with the text: 'Check out the deployment confidence dashboard and Sentry issues so you're the first to know if you've broken anything.' At the bottom is a URL: <https://github.blog/2021-01-25-improving-how-we-deploy-github/>.

# MVP

```
type: v1/appservice
services:
  app:
    image: app
    ports:
      - name: http
        port: 8080
    healthcheck:
      type: http
      httpPath: /api/healthy
      port: 8080
      interval: 10
      startPeriod: 30
    images:
      app:
        dockerfile: ./Dockerfile

routes:
  - name: http
    protocol: http
    host:
      - scope: internal
    backend:
      - service: app
        port: http
```



## 프로젝트 등록

어드민 내에서 구분할 프로젝트 이름과 GitHub 저장소의 주소를 입력해 주세요.

프로젝트 이름  
sre-test-server-demo

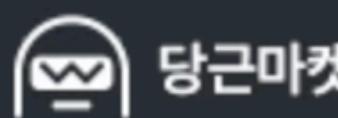
GitHub SSH 주소  
<https://github.com/daangn/sre-test-server-tmp>

Git 기본 브랜치  
main

**프로젝트 등록**

- Prepare 파이프라인을 생성
- Deploy Alpha KR 파이프라인을 생성
- Deploy Prod KR 파이프라인을 생성





당근마켓

프로젝트 등록 배포 미리보기

프로젝트 /sre-test-server-demo

# sre-test-server-demo

배포



## History



Wayland triggered 1-231dbbc with 231dbbc 22 minutes ago

상세 보기





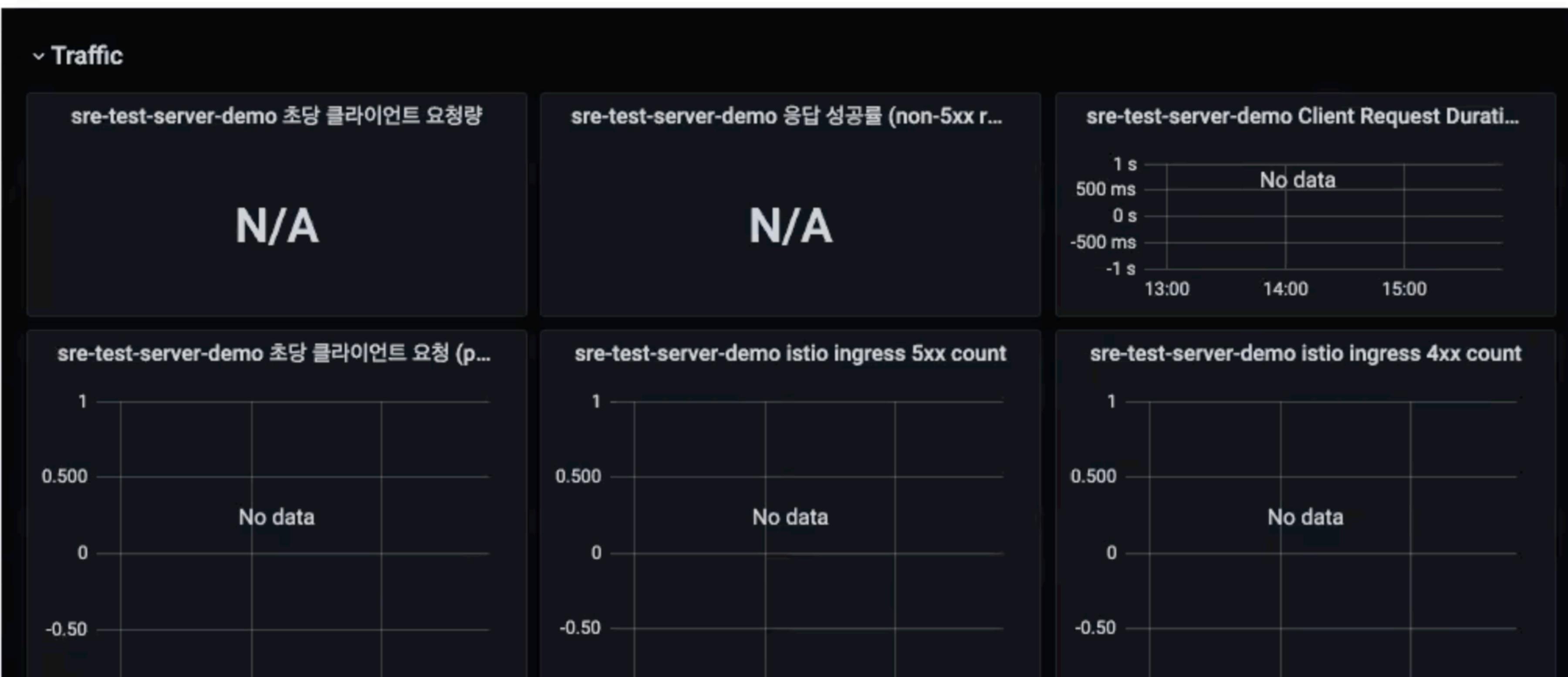
# sre-test-server-demo

배포

Deployment

monitoring

- [Grafana 대시보드](#)
- [Loki 로그](#)





프로젝트 /sre-test-server-demo /1-231dbbc

## 1-231dbbc

**"Prepare K8S" Passed**

Triggered by Wayland 29 minutes ago for 171 secs Commit 231dbbc

Logs

**"Deploy KR Alpha K8S" Passed**

26 minutes ago for 34 secs

Manifest

Logs

**"Deploy KR Prod K8S" Passed**

25 minutes ago for 15 secs

Deploy

Manifest

Logs

**IRSA, IAM**

```
type: v1/appservice

services:
  app:
    role: app-role

roles:
  app-role: .deploy/policies/app.json
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3>ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::test-service-bucket",
        "arn:aws:s3:::test-service-bucket/*"
      ]
    }
  ]
}
```

# Kontrol



SRE Delivery  
(Owen, Wayland)

**마이그레이션은 힘들다**



Kontrol을  
더 좋게 만든다

**배포 후 상태를 보고 싶다**



v2.3.0+f

APP DETAILS

APP DIFF

SYNC

SYNC STATUS

HISTORY AND ROLLBACK

DELETE

REFRESH ▾



Log out

APP HEALTH

Healthy

CURRENT SYNC STATUS

Synced

To 3.33.2 (3.33.2)

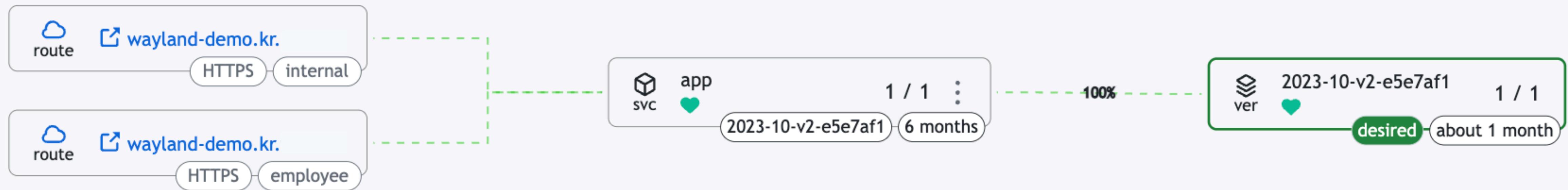
LAST SYNC RESULT

Sync OK

To 3.33.2

Succeeded a few seconds ago (Thu Feb 03 2022 11:48:36 GMT-0500)





[Summary](#)[K8S Spec](#)[AWS IAM Policy](#)

## SUMMARY

Name	wayland-demo-app
Class	k1.nano(cpu: 1 core, memory: 500Mi)
Image	app:e5e7af1aa9d83ac0abfc9e7b4cce2f39ec5ad304
Deploy Number	<a href="#">2023-10-v2</a>
Actions	<a href="#">Restart</a>

## SCALE

Replicas	1 / 1
HPA	Disabled

[Scale](#)[Enable](#)

## OBSERVABILITY

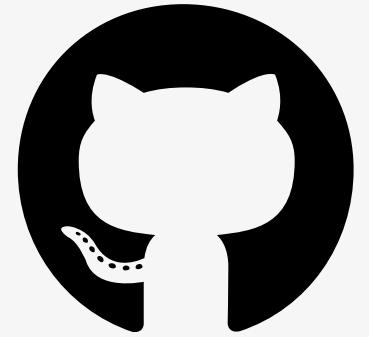
- [K8S Dashboard](#)
- [Service Dashboard](#)
- [Service Logs](#)

## INSTANCES

DEPLOY-NUM:2023-10-v2(1)

[Configs](#)[Resources](#)[Cost](#)[Settings](#)

# **파편화된 환경 변수**



**Repo for Argo CD**



**AWS Parameter Store**



**Configmap**



**External Secrets**





# AWS Secrets Manager

## Configs [Configs 적용 가이드](#)

[Config 생성하기](#)

서비스에서 사용할 환경변수를 관리합니다.

common



test



### 모든 서비스에 기본으로 제공되는 환경변수

**POD\_NAME** 파드의 이름

**POD\_NAMESPACE** 서비스가 배포된 네임스페이스(보통은 프로젝트 이름)

**SERVICE\_ACCOUNT** 서비스 어카운트

**POD\_IP** 파드에 할당된 IP

**HOST\_IP** 파드에 배포된 호스트 서버의 IP

**COUNTRY\_CODE** 배포된 국가코드(kr, ca, jp, gb 등)

**ENV** 배포된 환경(alpha, prod)

**DD\_AGENT\_HOST** Datadog 에이전트의 IP

**DD\_PORT** Datadog 에이전트의 포트

**DD\_ENV** Datadog에 제공하는 환경(alpha, prod)

# wayland-demo 🐧

[Deployments](#)[Workloads](#)[CronJobs](#)[Configs](#)[Resources](#)[Cost](#)[Settings](#)[← Configs](#)**common**

Alpha All ✘ Production All ✘ ⌂ ⚔️ 🍃

[새로고침](#)[저장](#)

환경 변수 이름	Alpha KR 🇰🇷	Production KR 🇰🇷	Alpha CA 🇨🇦	Production CA 🇨🇦	Alpha GB 🇬🇧	Production GB 🇬🇧
DB_HOST	***	Missing	Missing	Missing	Missing	Missing
DB_USERNAME	***	Missing	Missing	Missing	Missing	Missing
DEBUG	***	Missing	Missing	Missing	Missing	Missing
EXAMPLE_ENVIRONMENT_VARIABLE	Missing	***	***	***	***	***
	+ 추가					

# Full Cycle Developers at Netflix — Operate What You Build



Netflix Technology Blog · Following

Published in Netflix TechBlog · 10 min read · May 18, 2018



12.2K



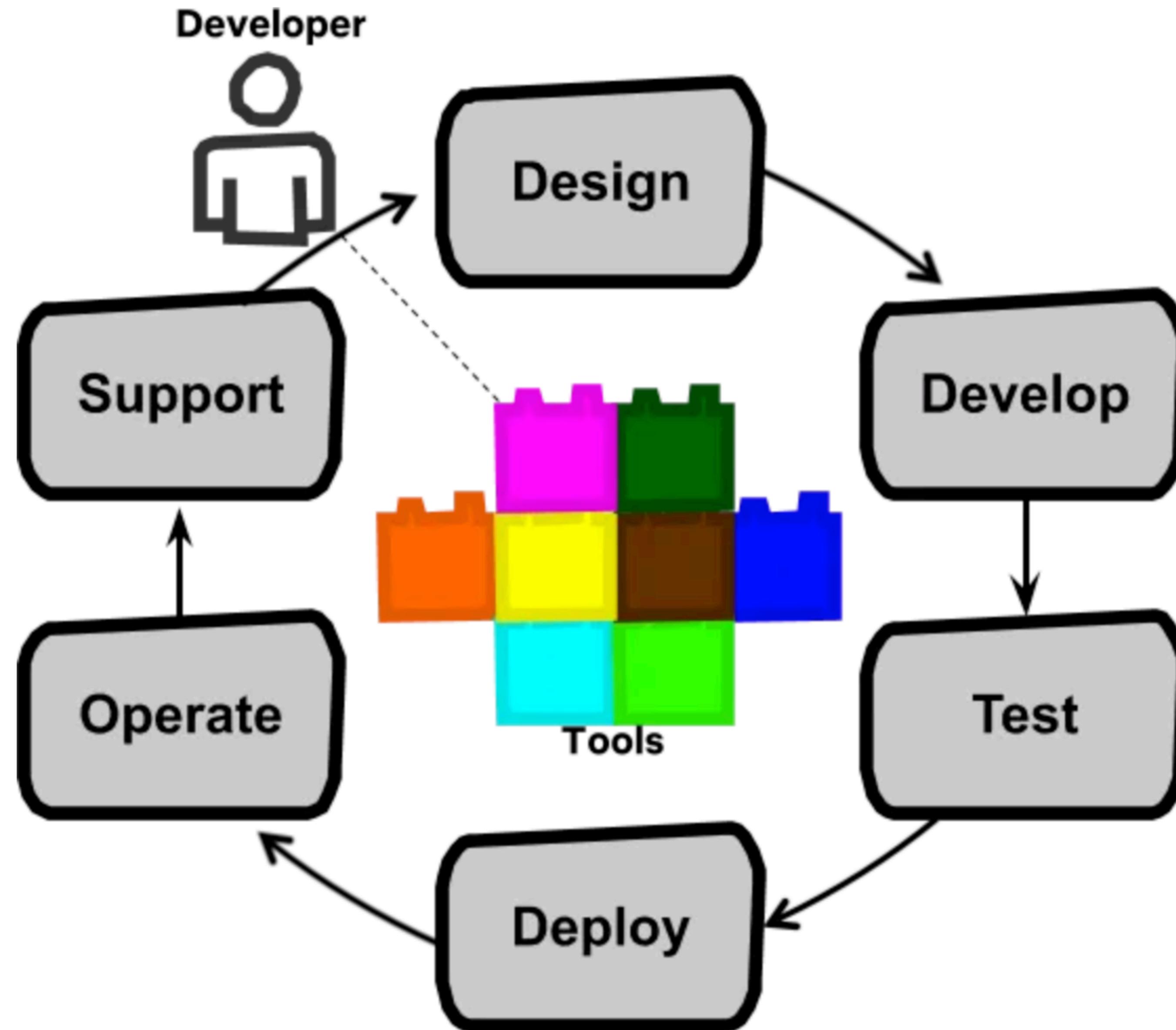
33



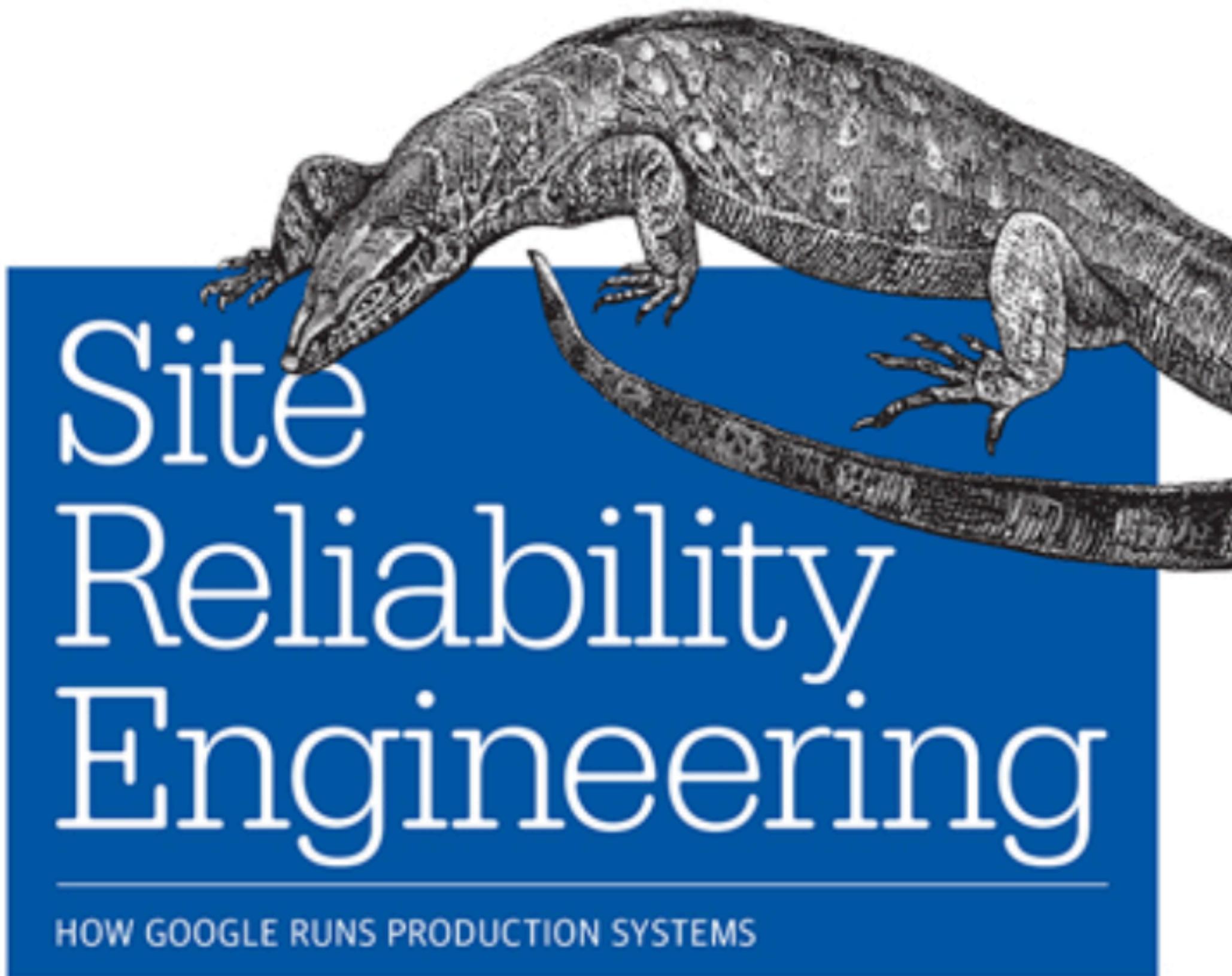
...

The year was 2012 and operating a critical service at Netflix was laborious. Deployments felt like walking through wet sand. Canarying was devolving into verifying endurance (“nothing broke after one week of canarying, let’s push it”) rather than correct functionality. Researching issues felt like bouncing a rubber ball between teams, hard to catch the root cause and harder yet to stop from bouncing between one another. All of these were signs that changes were needed.

Fast forward to 2018. Netflix has grown to 125M global members enjoying 140M+ hours of viewing per day. We’ve invested significantly in improving

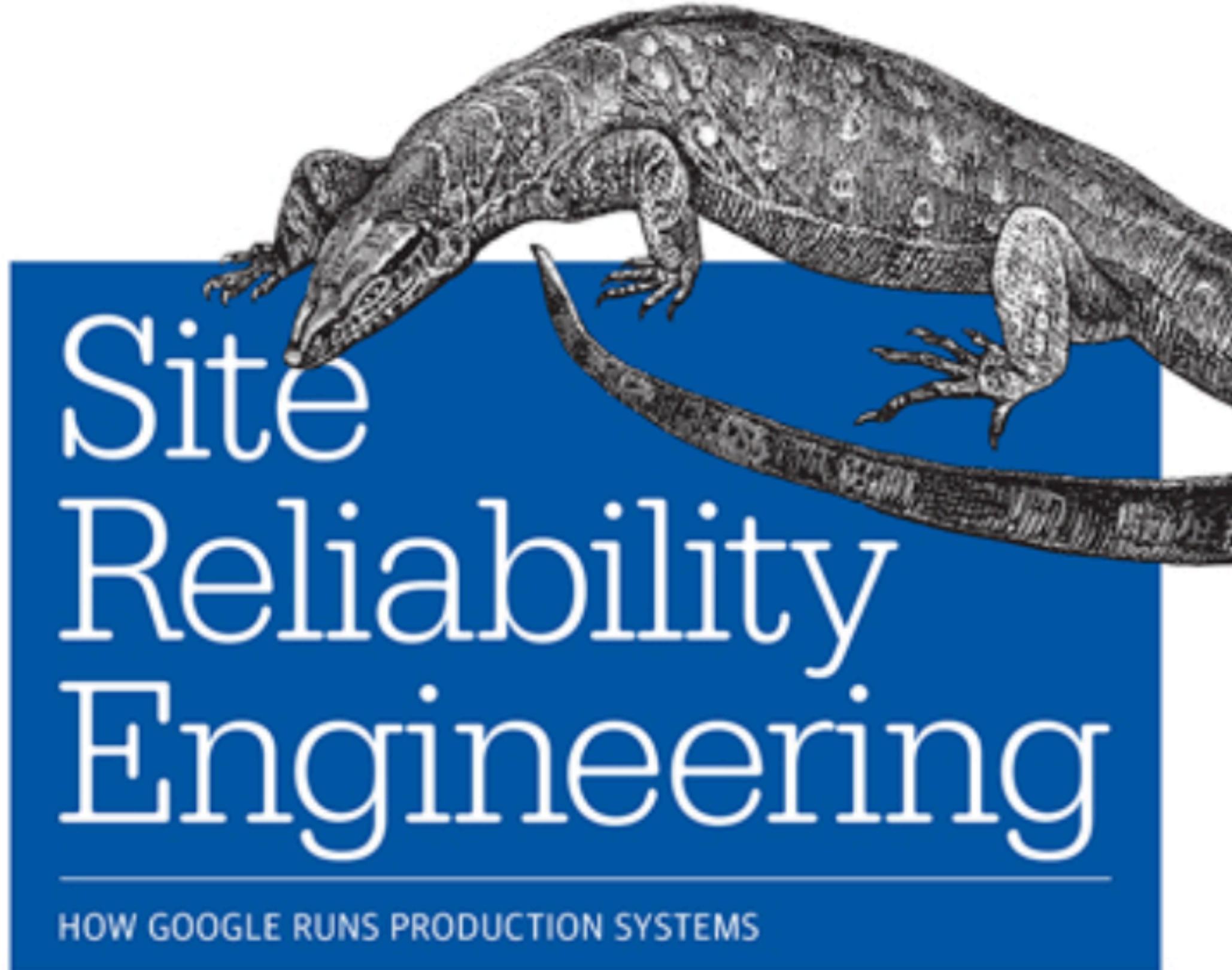


The empowered full cycle developer



Edited by Betsy Beyer, Chris Jones,  
Jennifer Petoff & Niall Murphy

“  
**SRE란 운영팀을 위한  
소프트웨어 엔지니어를 말한다.**  
”



“  
기본적으로 SRE팀은 엔지니어링에  
초점을 맞춘다는 점이 가장 중요하다.  
끊임없이 엔지니어링을 추구하지  
않으면 업무 부담이 증가하여  
그 부담을 나누기 위해  
더 많은 인력이 필요하게 되고  
결국에는 서비스의 크기에 따라  
전통적인 운영 업무를 담당하는 인력이  
기하급수적으로 늘어나게 된다.”  
”

Edited by Betsy Beyer, Chris Jones,  
Jennifer Petoff & Niall Murphy

<https://sre.google/sre-book/table-of-contents/>



“  
이러한 숙명에서 벗어나려면 서비스를  
관리하는 팀은 코드를 작성해야 한다.  
그렇지 못하면 늘어나는 일감에 파묻히게  
될 뿐이다.  
그래서 구글은 SRE팀에 티켓,  
전화 응대, 수작업 등,  
소위 '운영' 업무에 최대 50%의 시간만  
투입하도록 정해두고 있다.”

Edited by Betsy Beyer, Chris Jones,  
Jennifer Petoff & Niall Murphy

<https://sre.google/sre-book/table-of-contents/>

# **프로젝트 소유권 관리**

당근마켓 / 인프라실 / SRE 팀 / 딜리버리 파트

# wayland-demo 🐱

 Deployments

 Workloads

 CronJobs

 Configs

 Resources

 Cost

 Settings

# Deployments



## About

wayland의 테스트 프로젝트



↗️ [빌드/배포 통계](#)

## Team

담당 팀: 딜리버리 파트

슬랙 그룹: sre-delivery

슬랙 채널: [\\_infra-sre](#)

팀 문서: [딜리버리 파트 문서](#)



**Wayland** Leader  
Site Reliability Engineer

GitHub ID: [outsider-biz](#)

부서: 인프라실/SRE 팀/딜리버리 파트

하는 일: SRE 팀 딜리버리 파트 리더. 배포 시스템 구축 및 운영, kontrol 개발 및 운영

담당 프로젝트

- [wayland-test-server](#)
- [sean-test-server](#)
- [alfred-test-server1](#)
- [kontrol-migration-helper](#)
- [jvm-playground-server](#)
- [ow-test-server](#)

# Resources

[ⓘ 지원 리소스 목록](#)

- 프로젝트들이 소유한 AWS 리소스를 관리해요.

AWS

GCP

All env ▾

All region ▾

All typeName ▾



Create ▾

Resource search results (24)

[다른 프로젝트로 보내기](#)

Redis Production KR

[demo](#)

S3 Production KR

[Alpha](#)

[demo](#)

Redis Alpha KR

[demo-main](#)

S3 Production KR

[demo-static-resource](#)

S3 Alpha KR

[demo-static-resource-main](#)

# 전체 리소스

[지원 리소스 목록](#)

- 자신의 프로젝트의 리소스탭에서 찾을 수 없는 전체 리소스를 검색해 볼 수 있어요.
- 프로젝트에 할당 안되어 있는 리소스를 할당하거나 다른 프로젝트에 할당된 리소스를 찾을 수 있어요.

AWS

GCP

All env

All region

All typeName



미할당 리소스만 보기

On

aws arn을 입력해 주세요.

## Resource search results (685)

Page Size (50)

프로젝트 할당하기

CloudFront Distribution Production KR   
E16A9LSOVAGTRU

CloudFront Distribution Production KR   
E1I02RGZ8WZLR8(prod-demo1-webview)

CloudFront Distribution Production KR   
E22YPAOWL7VNXX

CloudFront Distribution Production KR   
E2HMNQ1NVJPPZ6(prod-local-demo2-webview-webview)

CloudFront Distribution Production KR   
E2QTY8HZD5HID3

CloudFront Distribution Production KR   
EA980QPV05O8K(production-demo3-center)

A photograph of a snowy urban street. In the foreground, a dark metal fence is completely covered in a thick layer of white snow. Several street lamps with yellow globes are visible, their light partially obscured by falling snow. The background shows more snow-covered trees and buildings, creating a soft, monochromatic winter atmosphere.

# IT 겨울

**비용의 가시성을 높여야 한다**

# wayland-demo

[Deployments](#) [Workloads](#) [CronJobs](#) [Configs](#) [Resources](#) [Cost](#) [Settings](#)

## Cost

프로젝트가 소유한 리소스들의 비용을 조회해요.



## Cost breakdown

Service	Service total	4 months trend	August 2023	September 2023	October 2023	November 2023	Dec
Total costs	\$526.11	█	\$336.31	\$86.24	-	\$30.23	

# 사용성이 좋지 않은 Kubernetes 크론잡



APP DETAILS

APP DIFF

SYNC

SYNC STATUS

HISTORY AND ROLLBACK

DELETE

REFRESH



## APP HEALTH

Healthy

## CURRENT SYNC STATUS

OutOfSync From release/prod (b8ca33f)

MORE

Author:

Comment:

## LAST SYNC RESULT

Sync OK

MORE

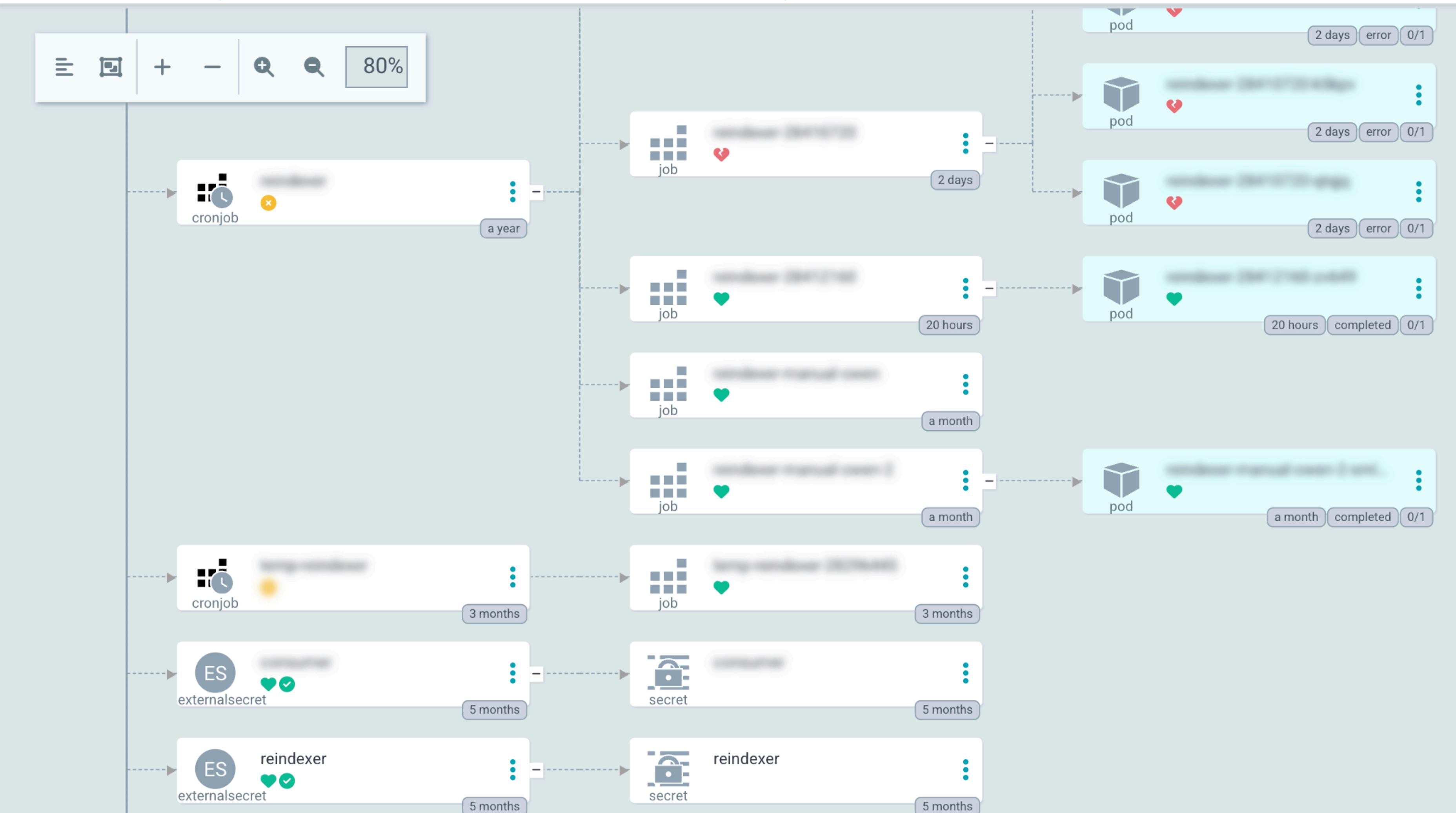
To 05875d5

Succeeded 6 days ago (Wed Jan 03 2024 18:40:45)

GMT+0900

Author:

Comment:



Applications

Settings

User Info

Documentation



## FILTERS

## NAME

NAME

## KINDS

KINDS

## SYNC STATUS

 Synced 35 OutOfSync 11

## HEALTH STATUS

 Healthy 60

 Deployments

 Workloads

 CronJobs

 Configs

 Resources

 Cost

 Settings

알파

 한국

프로덕션

 한국

**Production**

 크론 잡 목록

**cron**

스케줄:(\*/\* \* \* \*) "5분마다"

다음 시작 시간(UTC) :2024-01-08 18:10:00

[← CronJobs](#)**cron**[▶ 수동실행](#)[크론잡 비활성화](#)[Production](#) [잡 목록](#) cron-28412285  
prod-kr-cron

1분 미만 전 생성

[중지](#)[로그](#)[그라파나로 보기](#) cron-28161160  
prod-kr-cron

6개월 전 생성

5s

[삭제](#)[로그](#)[그라파나로 보기](#) cron-28161155  
prod-kr-cron

6개월 전 생성

5s

[삭제](#)[로그](#)[그라파나로 보기](#) cron-28161150  
prod-kr-cron

6개월 전 생성

4s

[삭제](#)[로그](#)[그라파나로 보기](#) cron-28161145  
prod-kr-cron

6개월 전 생성

4s

[삭제](#)[로그](#)[그라파나로 보기](#)

# **운영의 사용성 개선**

# wayland-demo

Deployments Workloads CronJobs Configs Resources Cost Settings

## Deployments

Alpha Production All X | ? 배포 Rollback

- Wayland scaled in(1 -> 0) app in Production 1 minute ago
- Wayland scaled out(0 -> 1) app in Production 3 minutes ago
- Wayland scaled in(10 -> 0) app in Production 27 days ago
- Wayland scaled in(1 -> 0) app in Alpha 27 days ago
- Wayland scaled out(0 -> 1) app in Alpha about 1 month ago
- Wayland updated config test 총 1개의 환경 변수 업데이트 in Production about 1 month ago
- Wayland updated config test 총 1개의 환경 변수 업데이트 in Alpha about 1 month ago
- Wayland updated config test 총 1개의 환경 변수 업데이트 in Production about 1 month ago
- Wayland updated config test 총 1개의 환경 변수 업데이트 in Alpha about 1 month ago
- Wayland updated config test 총 1개의 환경 변수 업데이트 in Production about 1 month ago

## About

프로젝트 설명이 없습니다.

[등록하기](#)

 [등록하기](#)

 [빌드/배포 통계](#)

## Team

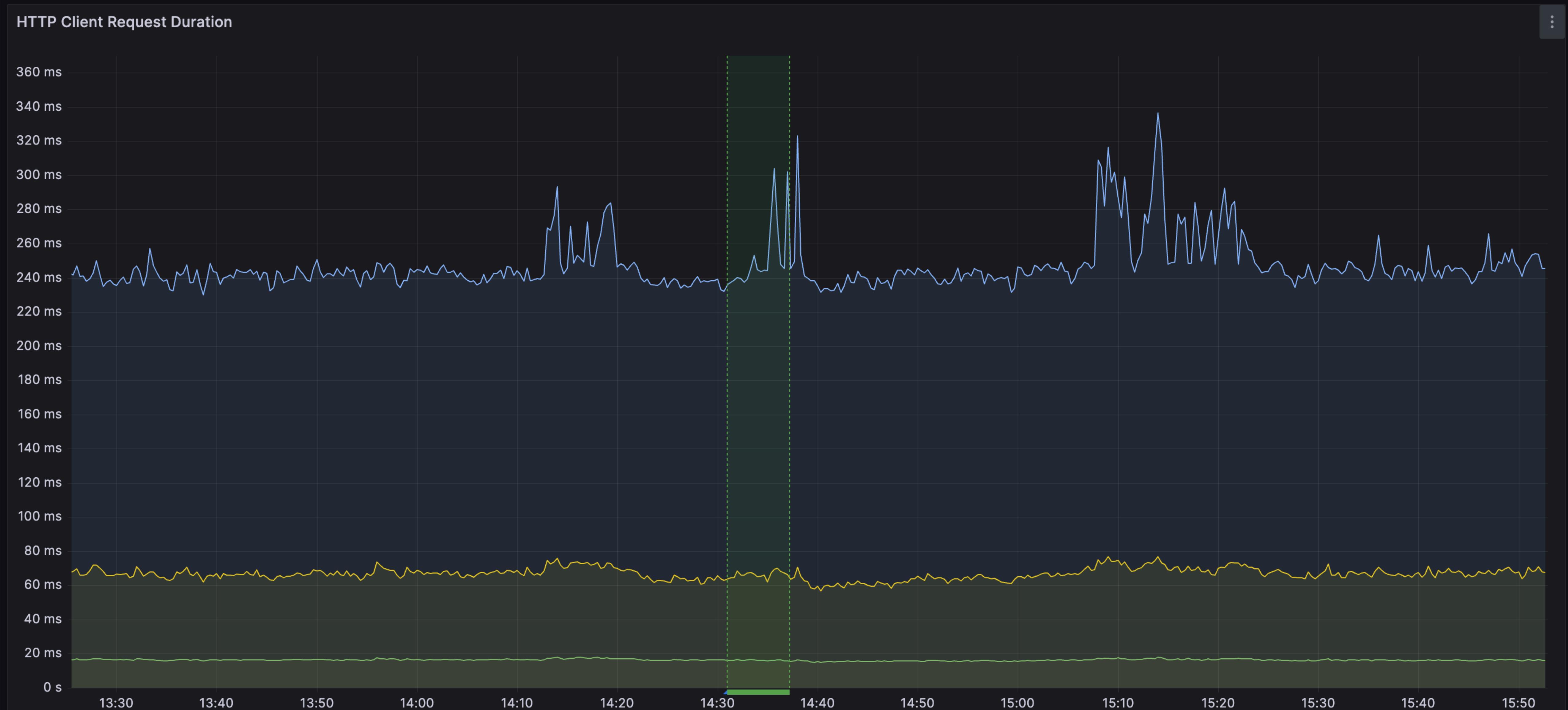
담당 팀: 딜리버리 파트

슬랙 그룹: sre-delivery

슬랙 채널: \_infra-sre

팀 문서: [딜리버리 파트 문서](#)





클라우드 리소스의 소유권을  
관리할 수 있으니  
생성도 하고 싶다

# wayland-demo

Deployments Workloads CronJobs Configs Resources Cost Settings

## Resources

ⓘ 지원 리소스 목록

- 프로젝트들이 소유한 AWS 리소스를 관리해요.

AWS

GCP

All env

All region

All typeName



Create

Activity

활동이 없어요.

Resource search results (0)

다른 프로젝트

No resources.

Redis

Static WebSite

DynamoDB Table

S3 Bucket

# wayland-demo

Deployments Workloads CronJobs Configs Resources Cost Settings

[← Resources](#)

## DynamoDB

DynamoDB 자원을 생성해요.

### 사용 목적 \*

테이블 이름을 결정하는데 사용됩니다.

- 최대 길이: 200자 미만.
- 키 이름: 영어 대/소문자, 숫자, '-', '\_', '!' 특수문자만 허용.

### 키 정의 \*

파티션 키 유형      파티션 키 이름 \*

String

정렬 키 유형      정렬 키 이름

String

테이블의 파티션 키와 정렬 키를 설정합니다.

- 정렬 키 이름에 아무 값도 없을 시, 정렬 키가 설정되지 않음.
- 키 이름: 영어 대/소문자, 숫자, '-', '\_', '!' 특수문자만 허용.

### 용량 모드 설정 \*

#### Provisioned with AutoScaling

- 사용율이 목표 사용율을 초과하면 읽기/쓰기 용량이  
 자동으로 늘어나요.  
 일반적인 경우 기본값(50%)를 사용하시면 충분해  
 요.

#### 목표 사용율(%) \*

기본값: 50

### 생성될 자원 리뷰

DynamoDB 생성 0

#### DynamoDB 테이블

#### 상세 옵션

##### 키 정의

##### 파티션 키

- 정의되지 않음

##### 정렬 키

- 사용하지 않음

용량 모드: Provisioned with AutoScaling

- 목표 사용율: 50

현재 사용 용량 / 최대 용량 = 50% 수준으로 Provisioned Capacity를  
 Kubernetes 합니다.

# EOF

 [outsideris](#)

 [outsideris](#)

 [outsideris@gmail.com](mailto:outsideris@gmail.com)