생산성 높은 MSA 환경 구성하기

Seongguk Choi, 2025. 8. 1

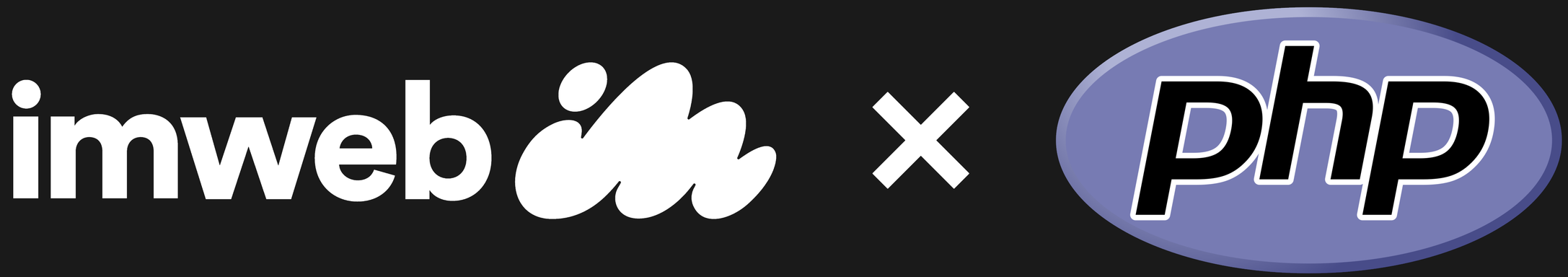
생산성(Productivity) 팀?

- 개발자들이 비즈니스 로직에만 집중할 수 있도록 개발 환경과 인프라를 개선
- 개발자의 개발 경험(DX)을 개발하는 팀

목차

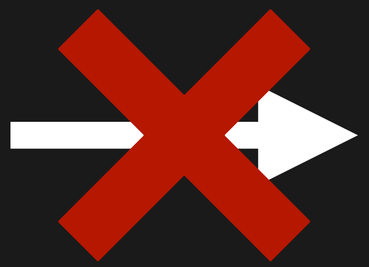
- 배경과 도전 과제
- MSA 전환 후 발생한 구체적인 문제
- 목표와 해결 방향
- 인프라 구성 자동화
- 인증 시스템 개선
- API 게이트웨이 도입
- 결과와 교훈
- Q&A

배경: 거대한 PHP 모놀리스

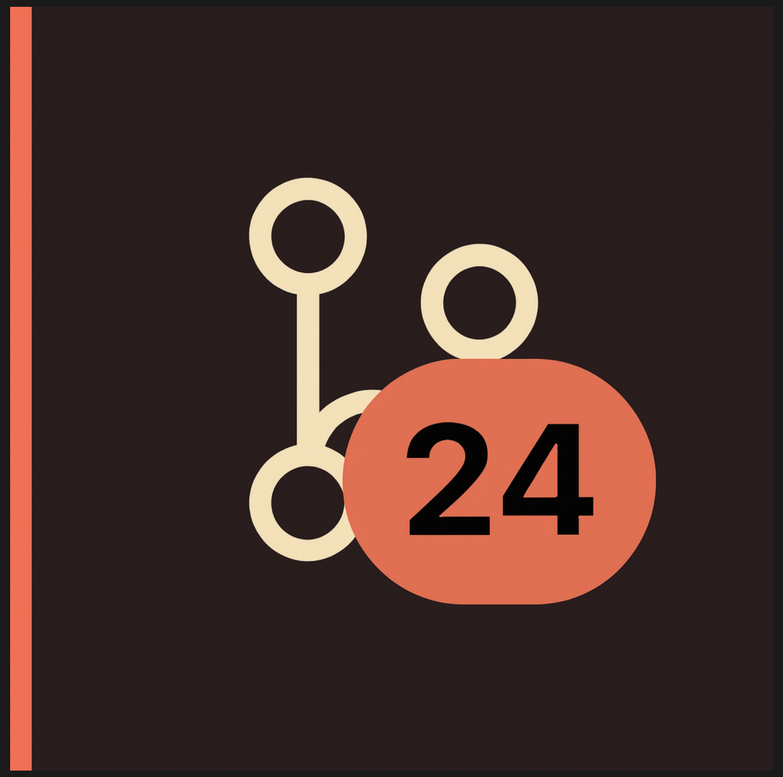


15 YEARS

배경: 거대한 PHP 모놀리스

7.0  8.0

배경: 거대한 PHP 모놀리스



배경: 거대한 PHP 모놀리스

server-bot APP 1:17 PM

test

attachment

```
0: 2 readfile(): SSL: Connection reset by peer -
/home/doznut/html/common/class/file_download.cls:304
1:
  0:
    file: /home/doznut/html/common/class/debug.cls
    line: 172
```

Show more

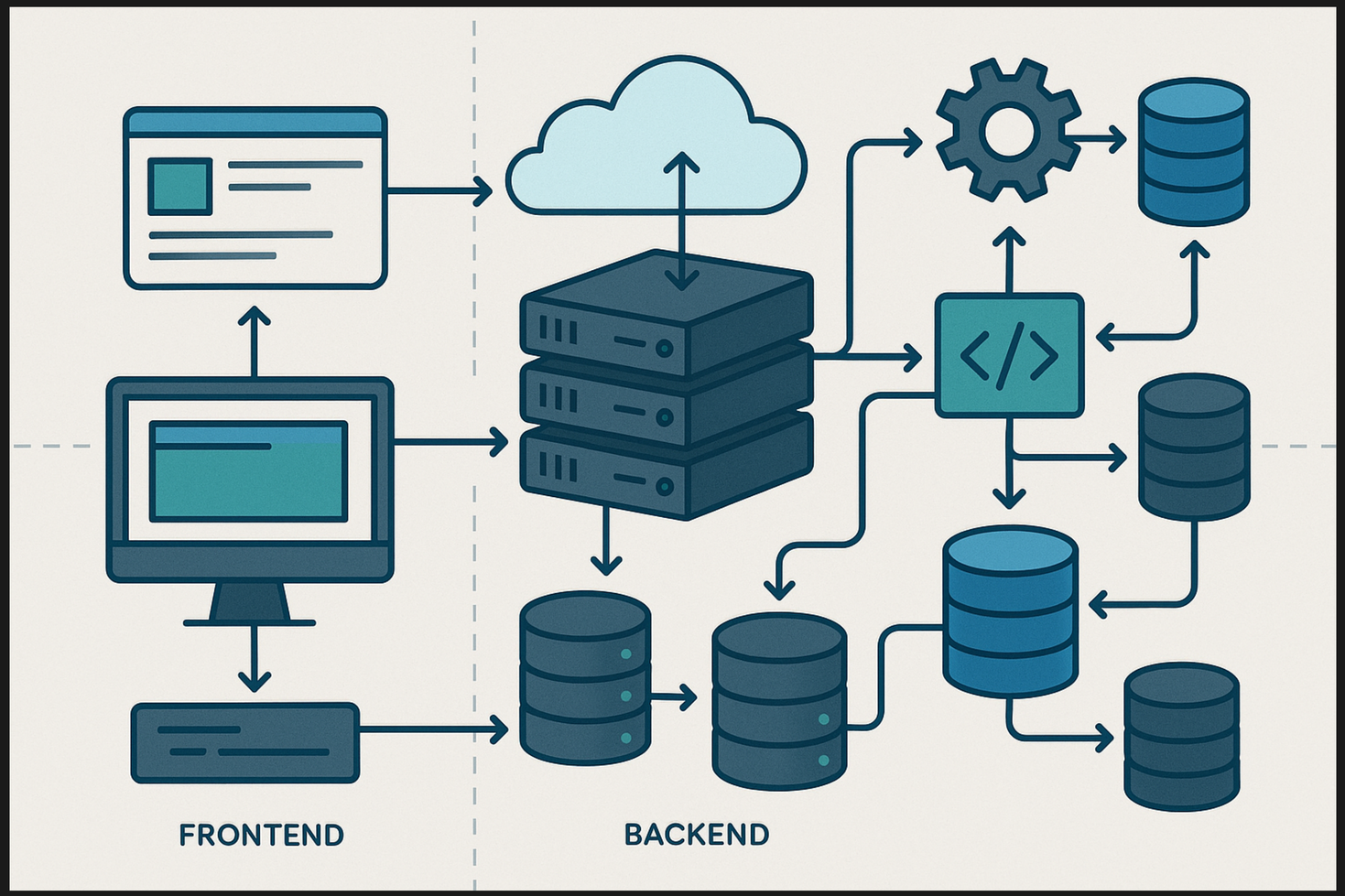
기대 vs 현실: MSA 도입의 함정

Repository

Deployments

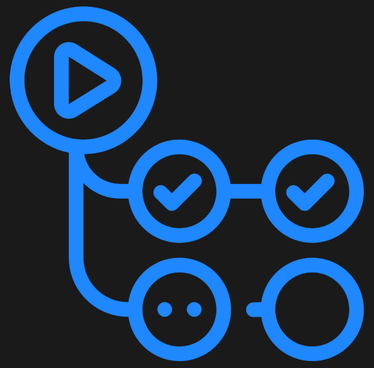
Feedback

기대 vs 현실: MSA 도입의 함정



기대 vs 현실: MSA 도입의 함정

ECR



GitHub Actions

Kubernetes

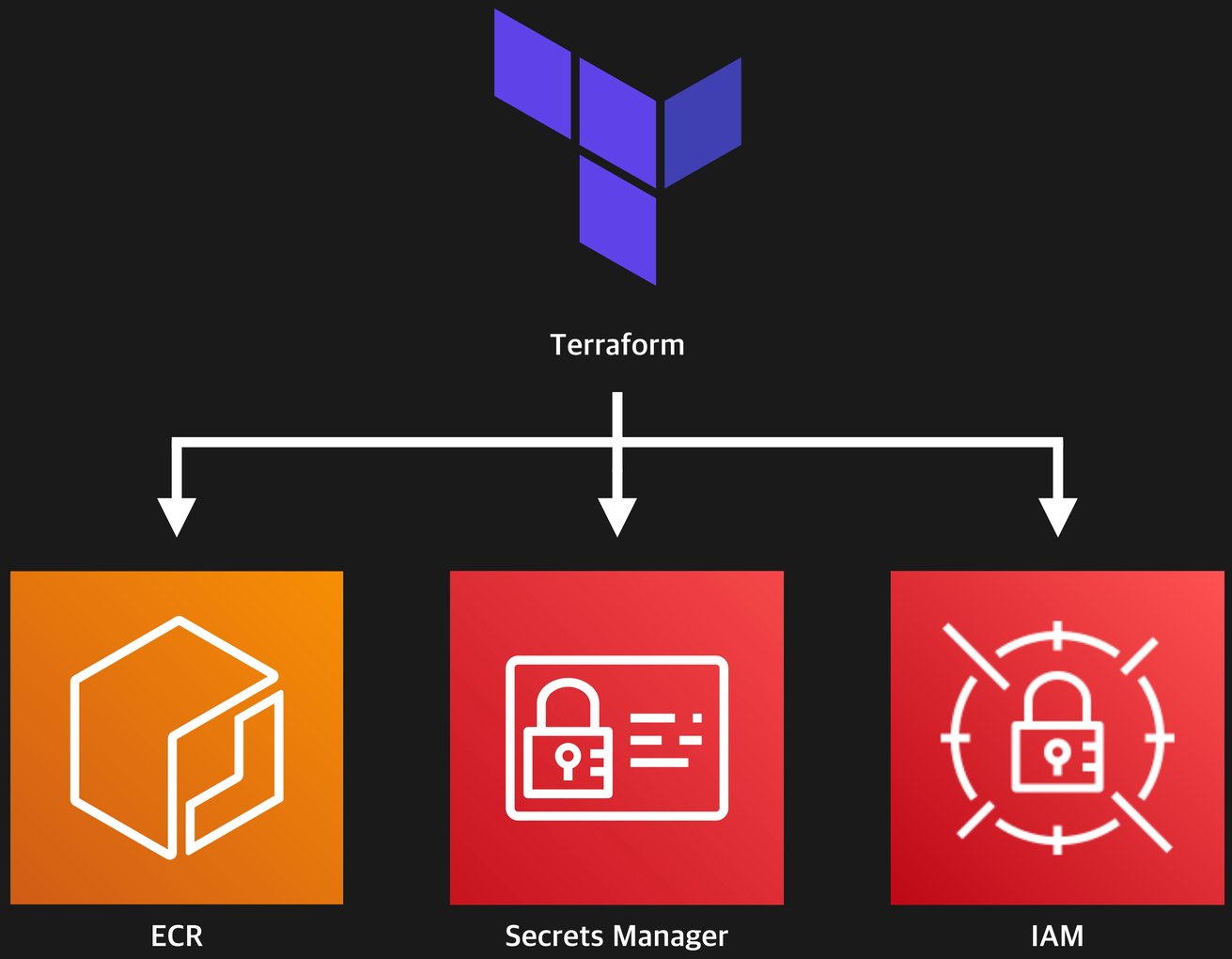
Secrets Manager

EKS

ArgoCD

MSA 전환 후 인프라 복잡성의 구체적 문제

분산된 인프라 코드 관리



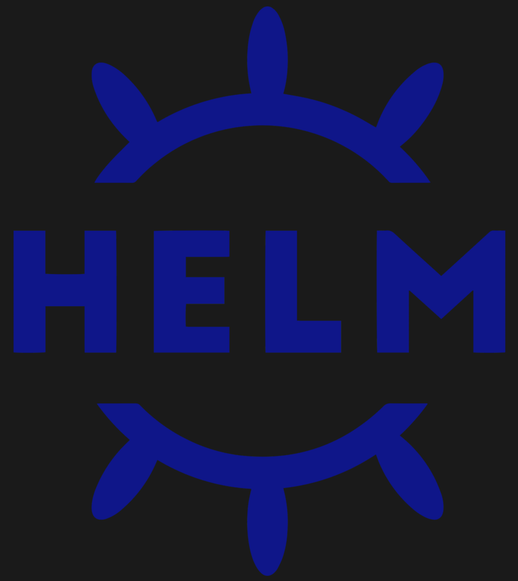
분산된 인프라 코드 관리

Kubernetes

Istio

분산된 인프라 코드 관리

ArgoCD



Helm

반복적인 수작업

인증 구현의 파편화

OAuth 2.0

Passport

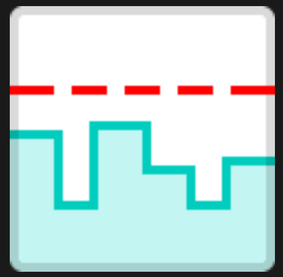
JWT

공통 기능의 중복 구현

Logging


Monitoring


CORS


Rate Limit

“수기 작업을 완전히 제거하자”

목표 설정

Reliability

Business

핵심 전략

Automation

Authentication

API Gateway

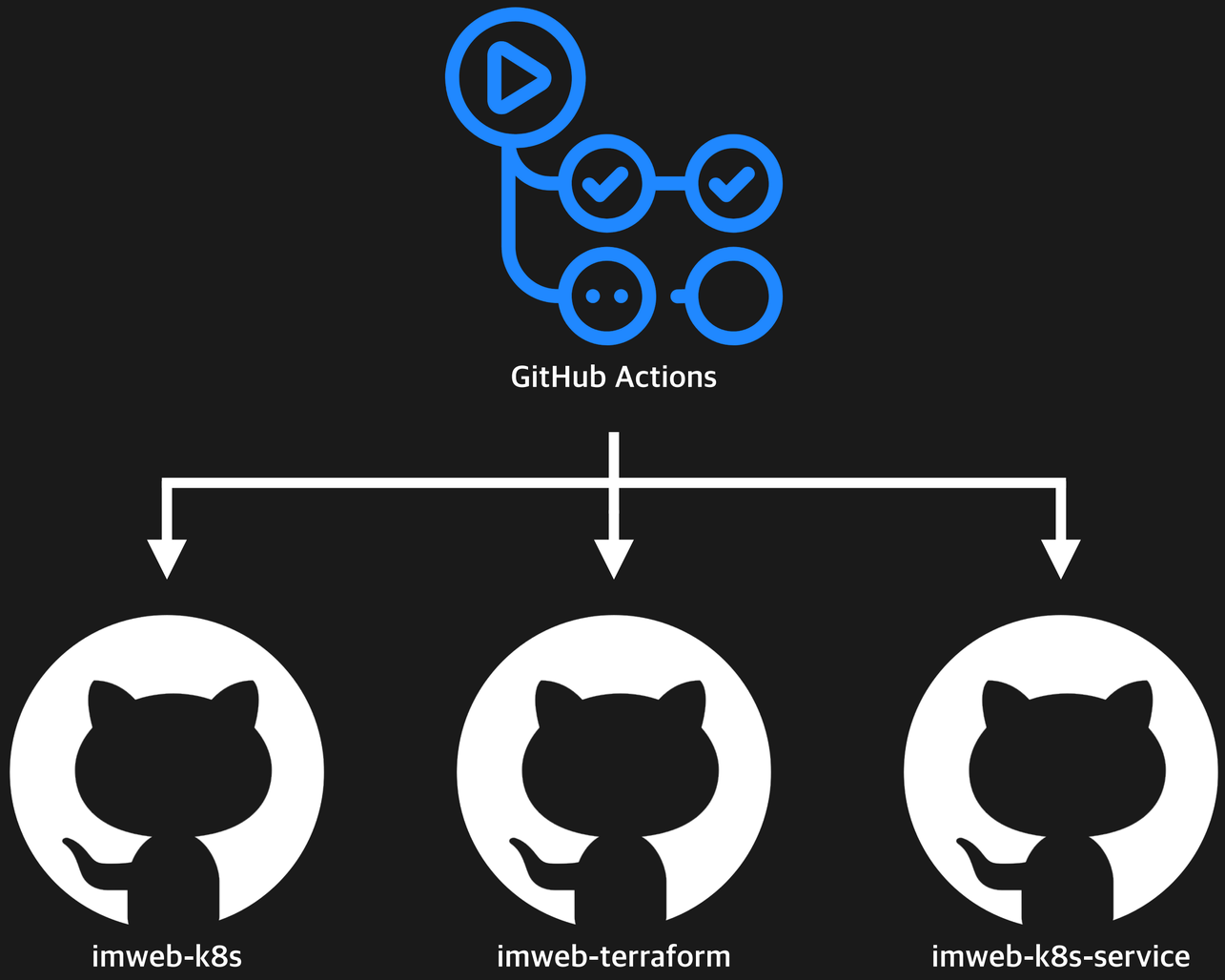
아임웹 인프라 설정

imweb-k8s

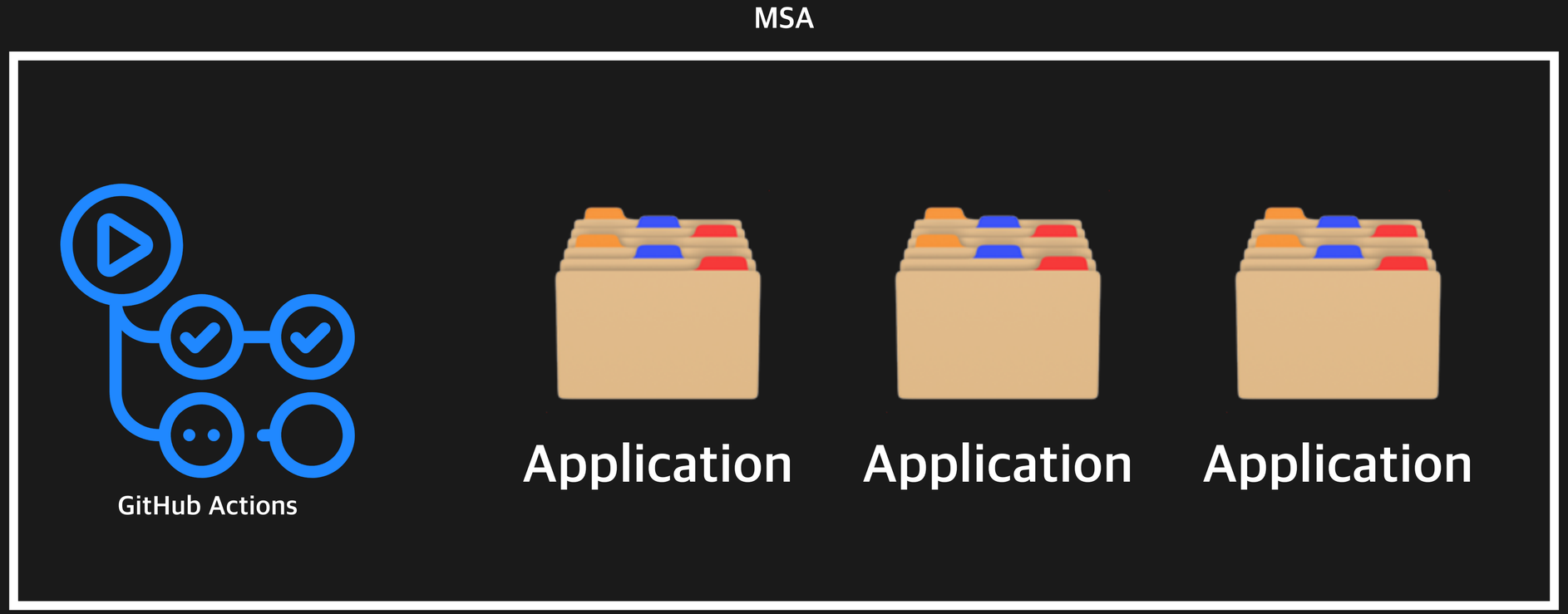
imweb-terraform

imweb-k8s-service

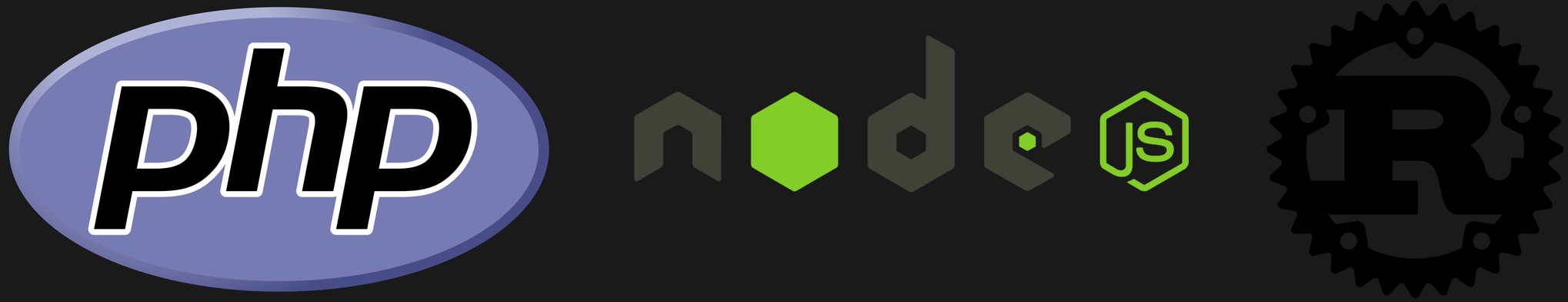
해결책 1: MSA 모노레포



해결책 1: MSA 모노레포



요구 사항



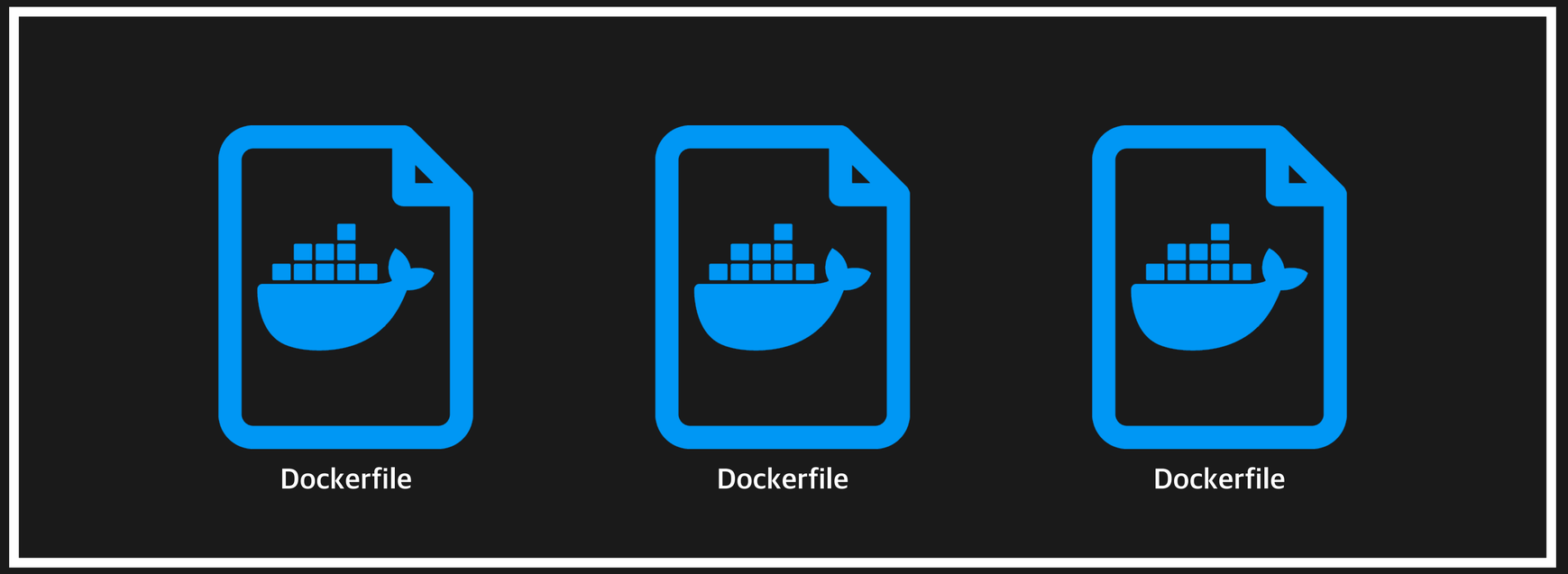
요구 사항

Deployment

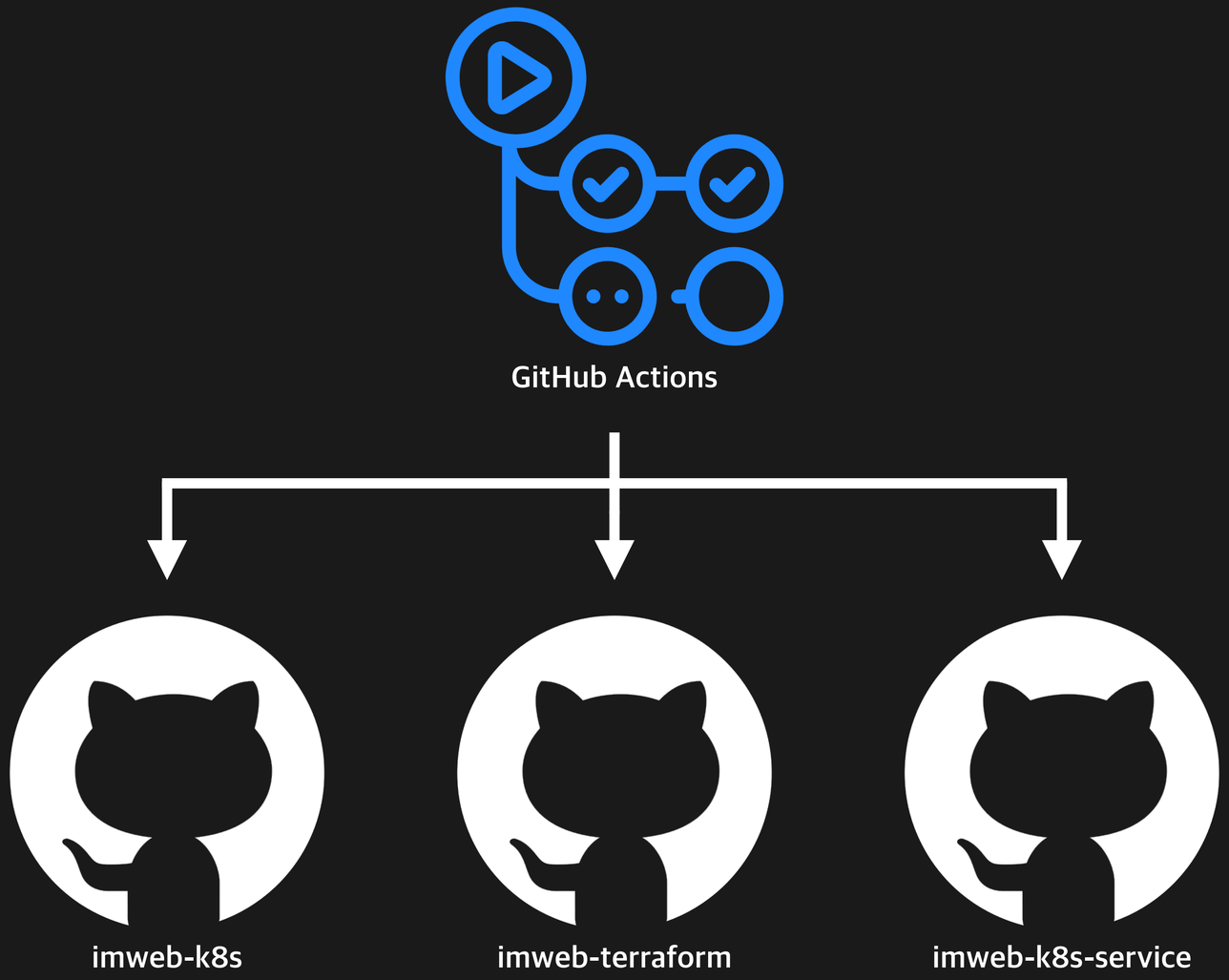
Kubernetes

최종안

/apps/{service}/{application}



최종안

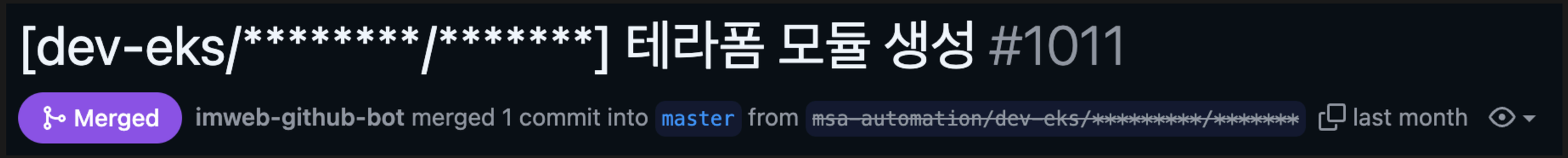

GitHub Actions
imweb-k8s
imweb-terraform
imweb-k8s-service

해결책 2: 사전 구성된 Terraform 모듈

해결책 2: 사전 구성된 Terraform 모듈

```
module "my-new-service" {
  source = "../modules/msa-application"
  name   = "my-new-service"
}
```

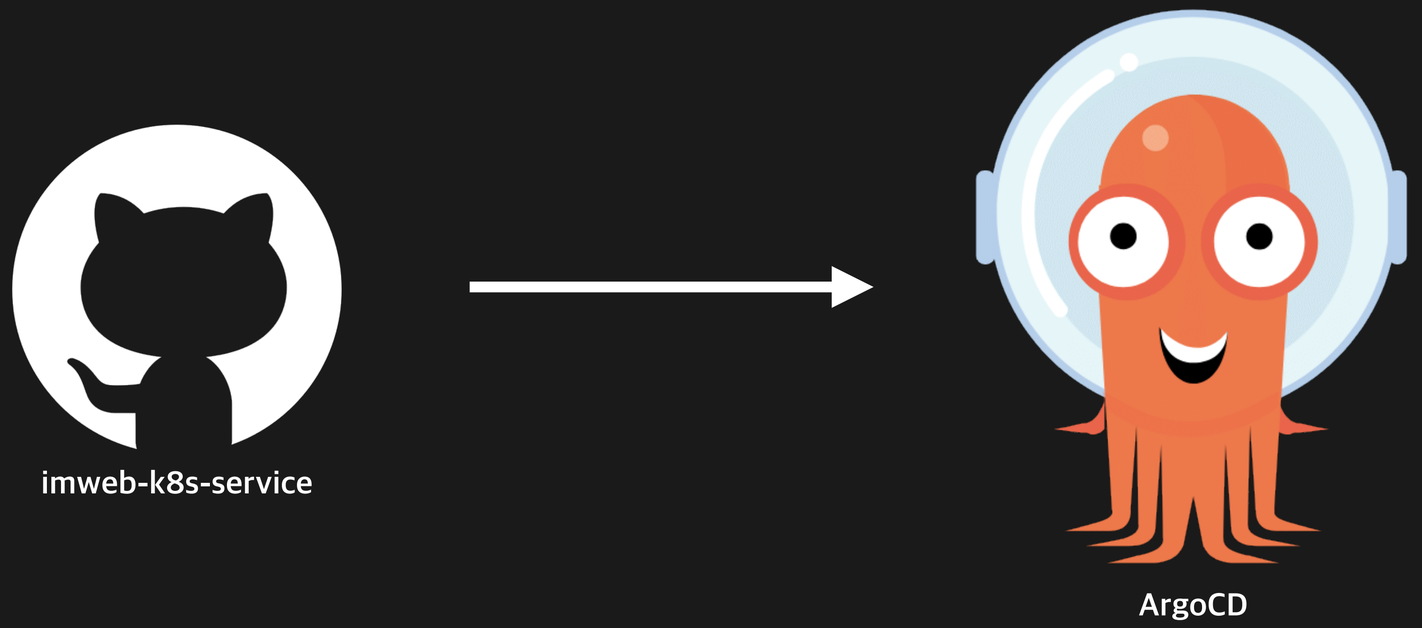
Terraform(Atlantis) 적용 자동화



Terraform(Atlantis) 적용 자동화

Terraform(Atlantis) 적용 자동화

ArgoCD 자동화



정리

1 Week

↓

20 Min.

-99%

인증 시스템 한계와 요구사항

PHP Session Redis

OAuth 2.0

"마이크로서비스 인증 흐름을 자동화하고, 클라이언트 SDK와 서버 모듈로 제공하자”

해결책 3: JWT 기반 인증 서버와 SDK/모듈

```
{
  "sub": "1234567890",
  "iat": 1516239022,
  "siteCode": "S0000000000000000",
  "unitCode": "u0000000000000000",
  "memberCode": "m0000000000000000",
  "role": "member"
}
```

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwiaWF0Ij
oxNTE2MjM5MDIyLCJzaXRlQ29kZSI6IlMwMDAwMDAwMDAwMDAwMDAwMCIsInVuaXRDb2RlI
joidTAwMDAwMDAwMDAwMDAwMDAwIiwibWVtYmVyQ29kZSI6Im0wMDAwMDAwMDAwMDAwMDAw
MCIsInJvbGUiOiJtZW1iZXIifQ.zQJVYXNSxRc1y5egiET2nGajKIJ4Q0R608p3NF24ba8
```

해결책 3: JWT 기반 인증 서버와 SDK/모듈

Auth Service

PHP Session Redis

해결책 3: JWT 기반 인증 서버와 SDK/모듈

해결책 3: JWT 기반 인증 서버와 SDK/모듈

```
@SiteAuth({ allowRoles: [...] })  // 인증 검증 데코레이터
@Get('profile')
getProfile(
  @SiteCode() siteCode: SiteCode,  // 사이트 코드 추출
  @MemberCode() memberCode: SiteMemberCode,  // 사용자 코드 추출
) {
  ...
}
```

API Gateway 필요성

Auth Service

API Gateway

API Gateway 옵션 비교

NGINX

Caddy®

Istio

- 쿠버네티스 환경에서 서비스 간 통신을 관리하는 서비스 메시
- 학습 곡선이 가파름
- 쿠버네티스 클러스터 외부의 서비스는 연결할 수 없다는 단점
- 아직 쿠버네티스 밖에 남아 있는 서비스까지 연결해야 하는 요구 사항

nginx/caddy

- 웹 서버이자 리버스 프록시
- 설정 유지보수가 복잡함
- 백엔드 엔지니어들이 각 플랫폼의 문법을 학습해야 하는 문제

Kong Gateway

- DB-less 모드로 YAML 기반의 설정이 가능
- "선언형 설정"이 가능하다는 점
- Kong이 설정을 데이터베이스가 아닌 파일로 관리
- Git을 통해 설정 형상 관리가 가능

선언형 설정?

```yaml
services:
  - name: my-service
    url: http://my-service.default.svc.cluster.local
    routes:
      - name: my-route
        paths:
          - /api/v1/users
        plugins:
          - name: site-auth  # JWT 검증 플러그인
```

결과 & 성과

- 현재 약 20여개의 마이크로서비스가 운영 중
- 매주 새로운 애플리케이션이 추가

결과 & 성과

2 Weeks

↓

1 Day

-93%

결과 & 성과

1 Week

↓

20 Min.

-99%

결과 & 성과

-98%

결과 & 성과

- 분산된 인프라 코드 → 단일 푸시로 자동화
- 반복적 수작업 → 템플릿화로 실수 제로
- 파편화된 인증 → 표준 SDK/모듈로 통일
- 중복 구현 → API Gateway로 중앙화

교훈 & 베스트 프랙티스

- "왜 느린가?"에 대한 근본 원인 분석이 핵심
- 모든 것을 한 번에 바꾸려고 하면 안 됨
- 시스템을 구성하면 애플리케이션별 중복 작업을 제거
- 개발자 경험(DX)에 투자하는 것은 비즈니스 가치 향상

결론

- 자동화와 표준화 없는 MSA는 복잡도 증가
- MSA는 목적이 아닌 조직 생산성의 도구
- **레거시를 MSA로 전환하는 것 자체가 목표가 되어서는 안 됨**

Q&A