
11

Feature Flag,
안전하고 신뢰 가능한 배포로 나아가는 열쇠



CONTENTS

Feature Flag?

- Feature Flag?
- Types of Feature Flag
- Why Feature Flag?

Feature Flag on Cloud-Native ecosystem

- OpenFeature & Flagd
- Deployment
 - Kubernetes-native
 - VM + K8s
- Flag Configuration
- Lab

Feature Flag?

“특정 기능을 동적으로 토글하기 위한 조건부 코드 실행 메커니즘”

“특정 기능을 동적으로 토글하기 위한 조건부 코드 실행 메커니즘”

```
Feature Flag Pseudo-code

1  FlagConfiguration flagConfiguration = FlagConfiguration.getInstance();
2  FeatureFlag flag = flagConfiguration.getFlag("FLAG_KEY");
3
4  boolean isFeatureEnabled = flag.getBooleanValue();
5
6  if (isFeatureEnabled) {
7      return RevealFeature();
8  }
9
10 return HideFeature();
```

Flag 정보 (Flag Configuration)

기능의 Routing을 결정 (Flag Router)

서로 다른 (기존/신규) 기능의 구분점 (Flag Point)

“특정 기능을 동적으로 토글하기 위한 조건부 코드 실행 메커니즘”

```
Feature Flag Pseudo-code

1  FlagConfiguration flagConfiguration = FlagConfiguration.getInstance();
2  FeatureFlag flag = flagConfiguration.getFlag("FLAG_KEY");
3
4  boolean isFeatureEnabled = flag.getBooleanValue();
5
6  if (isFeatureEnabled) {
7      return RevealFeature();
8  }
9
10 return HideFeature();
```

Flag 정보 (Flag Configuration)

기능의 Routing을 결정 (Flag Router)

서로 다른 (기존/신규) 기능의 구분점 (Flag Point)

“특정 기능을 동적으로 토글하기 위한 조건부 코드 실행 메커니즘”

```
Feature Flag Pseudo-code

1 FlagConfiguration flagConfiguration = FlagConfiguration.getInstance();
2 FeatureFlag flag = flagConfiguration.getFlag("FLAG_KEY");
3
4 boolean isFeatureEnabled = flag.getBooleanValue();
5
6 if (isFeatureEnabled) {
7     return RevealFeature();
8 }
9
10 return HideFeature();
```

Flag 정보 (Flag Configuration)

기능의 Routing을 결정 (Flag Router)

서로 다른 (기존/신규) 기능의 구분점 (Flag Point)

핵심은 코드를 수정하지 않고도 🙄🙄 시스템의 동작을 바꿀 수 있는 것

Types of Feature Flag

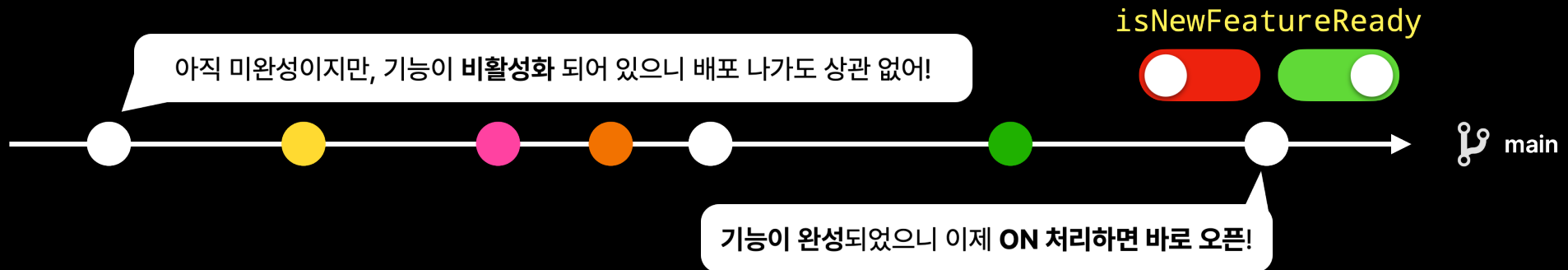
Release Toggle
Ops Toggle
Permission Toggle
Experimental Toggle

* Feature Flag = Feature Toggle, Feature Bits, Feature Flippers

Release Toggle

불완전한 코드임에도 불구하고 활성화되지 않은 상태로 Production 배포 가능

```
Feature featureA() {  
  if (isNewFeatureReady) {  
    return notReadyFeature();  
  }  
  
  return LegacyFeature();  
}
```



Ops Toggle

11 TechTalk 2023

시스템 동작의 운영 측면 제어



쇼킹딜

베스트

쿠폰/기획전

장보기

T공식대리점



아마존

N

아마존 딜

아마존 베스트

전지원님



###,###,###

바로가기 ON



쇼킹딜

베스트

쿠폰/기획전

장보기

T공식대리점



아마존

N

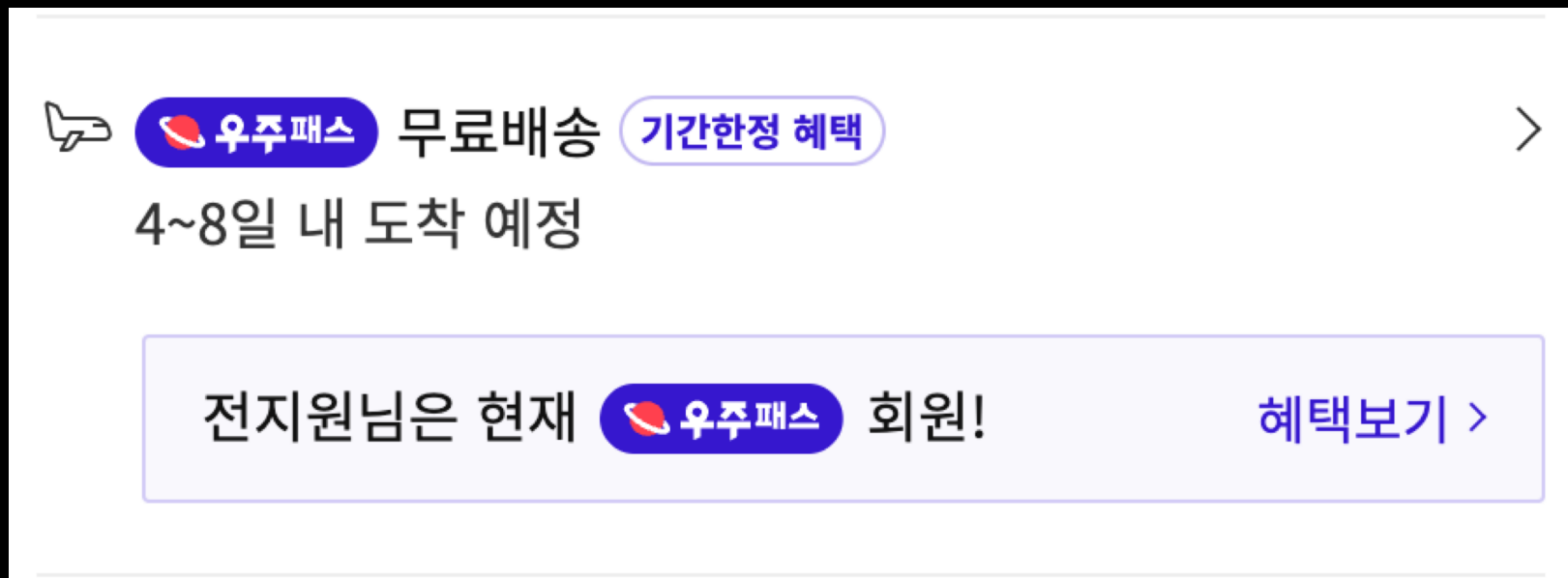
아마존 딜

아마존 베스트

Permission Toggle

11 TechTalk 2023

특정 사용자군에게만 노출



The screenshot displays a notification card with a white background and rounded corners. At the top left is a grey airplane icon. To its right is a blue pill-shaped button containing a white satellite icon and the text '우주패스'. This is followed by the text '무료배송' and another blue pill-shaped button with a white satellite icon and the text '기간한정 혜택'. A grey chevron icon is on the far right. Below this row is the text '4~8일 내 도착 예정'. A light grey rectangular box is positioned below the notification, containing the text '전지원님은 현재' followed by a blue pill-shaped button with a white satellite icon and the text '우주패스', and then '회원!'. To the right of this box is the text '혜택보기 >'.

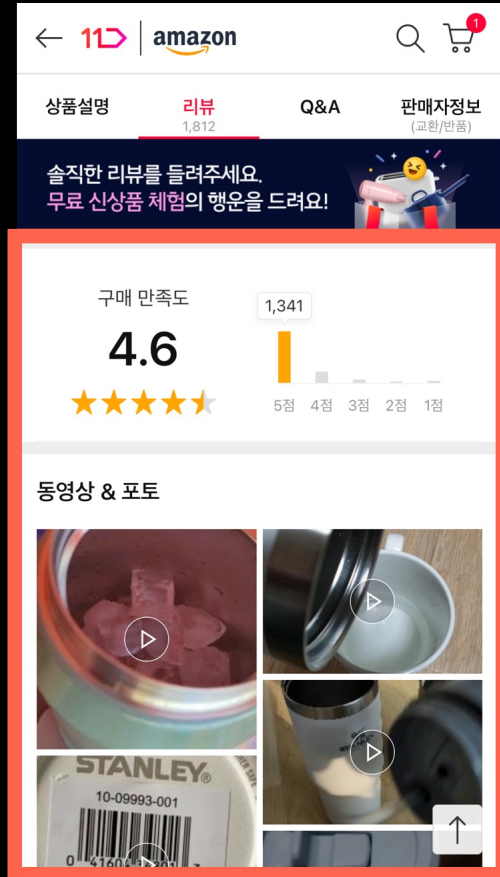
✈️ 우주패스 무료배송 기간한정 혜택 >

4~8일 내 도착 예정

전지원님은 현재 우주패스 회원! 혜택보기 >

Experimental Toggle

기능의 실험적 도입



상품

HTTP



리뷰



향상된 코드 협업

아직 사용자에게 노출되면 안되는 동료의 코드가 mainline에 있다보니
제가 작업한 기능을 노출하기 위해서는 기다려야만 했는데, 이제 그럴 필요가 없어요!

불필요한 배포, Production에서의 빠른 Feature 전환

"서비스 부하가 갑자기 높아져서 Slow Query가 요구되는 기능을 OFF했어요!"

"검증 환경에서는 정상이긴 했는데, 실제 서비스 환경에서 에러날까봐 걱정했어요. 😓 이제는 문제가 생기더라도 빠르게 전환할 수 있겠어요! 😊"

Context 정보를 조건으로 하는 조건부 기능 적용

"신규 기능을 우선 Host 1대에만 적용해보고, 이상이 없으면 전체 Host에 적용할 수 있더라구요!"

"T우주 구독상품을 이용 중인 특정 사용자에게만 배송비 무료 등의 기능을 적용할 수 있겠더라구요!"



서로 다른 구현, 관리

—> 사용현황 및 히스토리 파악 어려움

“공통의 표준”이 필요해!

Feature Flag on Cloud-Native ecosystem



Istio

Simplify observability, traffic management, security, policy with the Istio service mesh.



argo

Kubernetes-native tools to run workflows, manage clusters and do GitOps right



**CLOUD NATIVE
COMPUTING FOUNDATION**

**Open
Feature**

Standardizing Feature Flagging

CNCF Sandbox Project

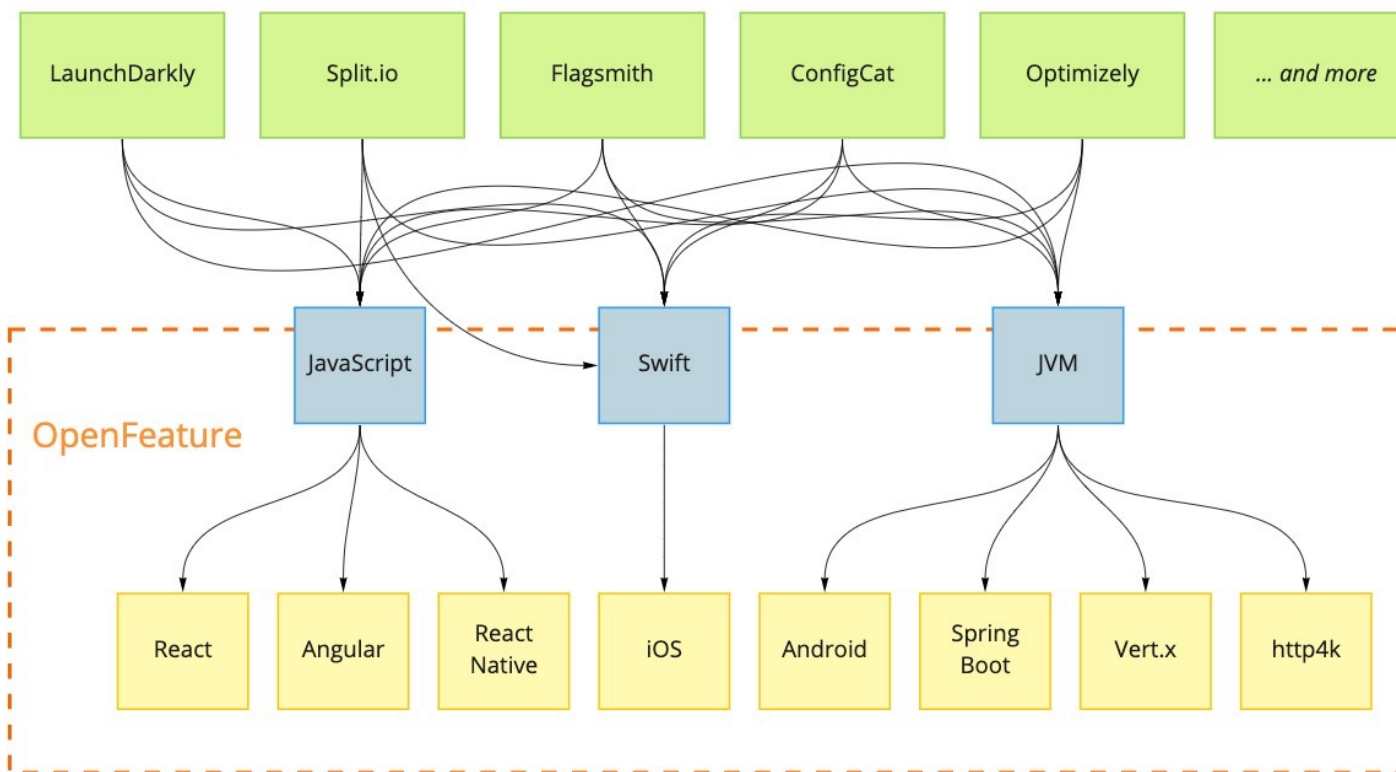


The Kubernetes Package Manager

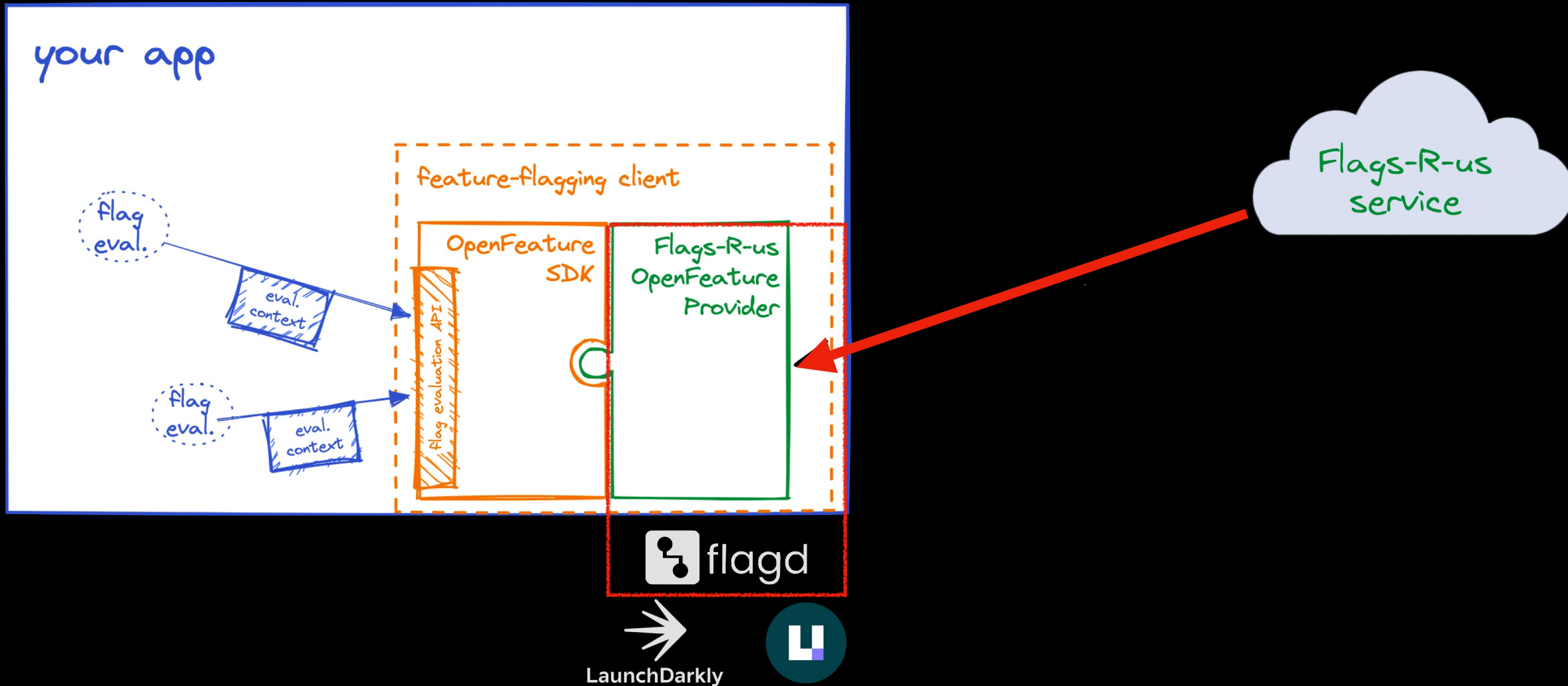


Thanos

Highly available Prometheus setup with long term storage capabilities.

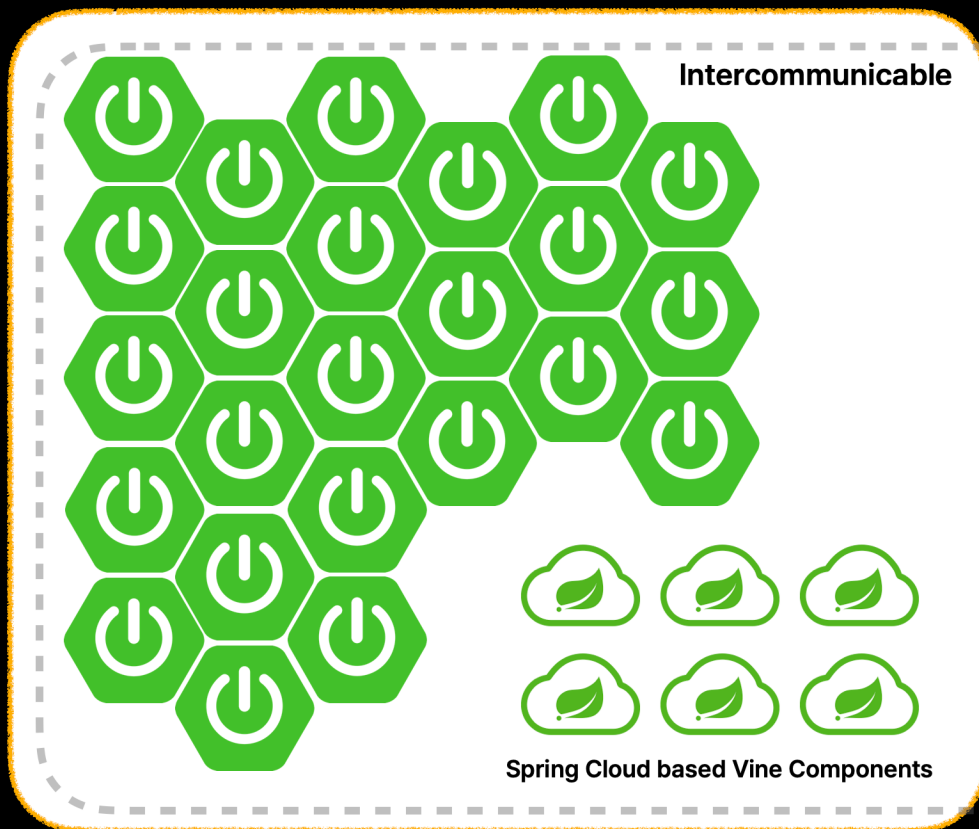


<https://openfeature.dev/blog/openfeature-a-standard-for-feature-flagging/>

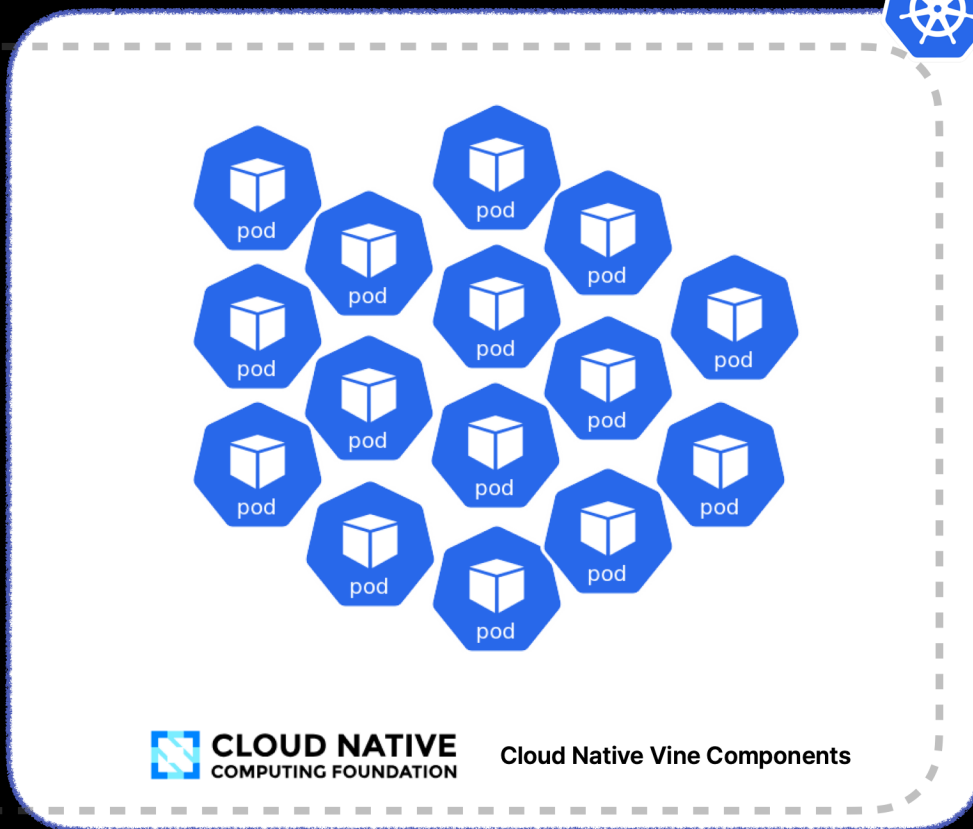


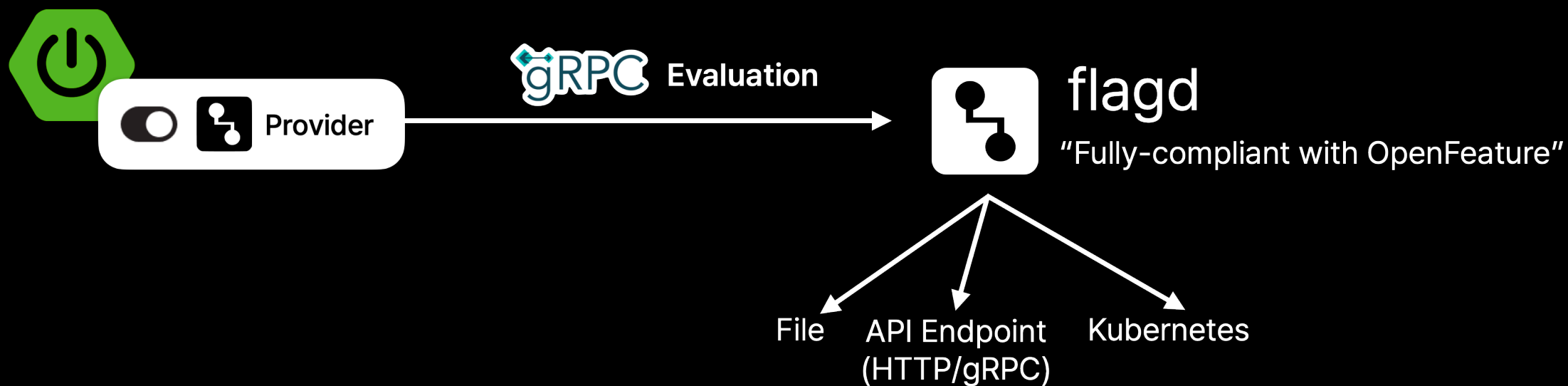
Vine Platform

IDC



AWS EKS





Deployment

Kubernetes-native (Operator)



INJECT!



```
annotations:  
  openfeature.dev/enabled: true  
  openfeature.dev/featureflagsource: "[namespace]/[crName],..."
```

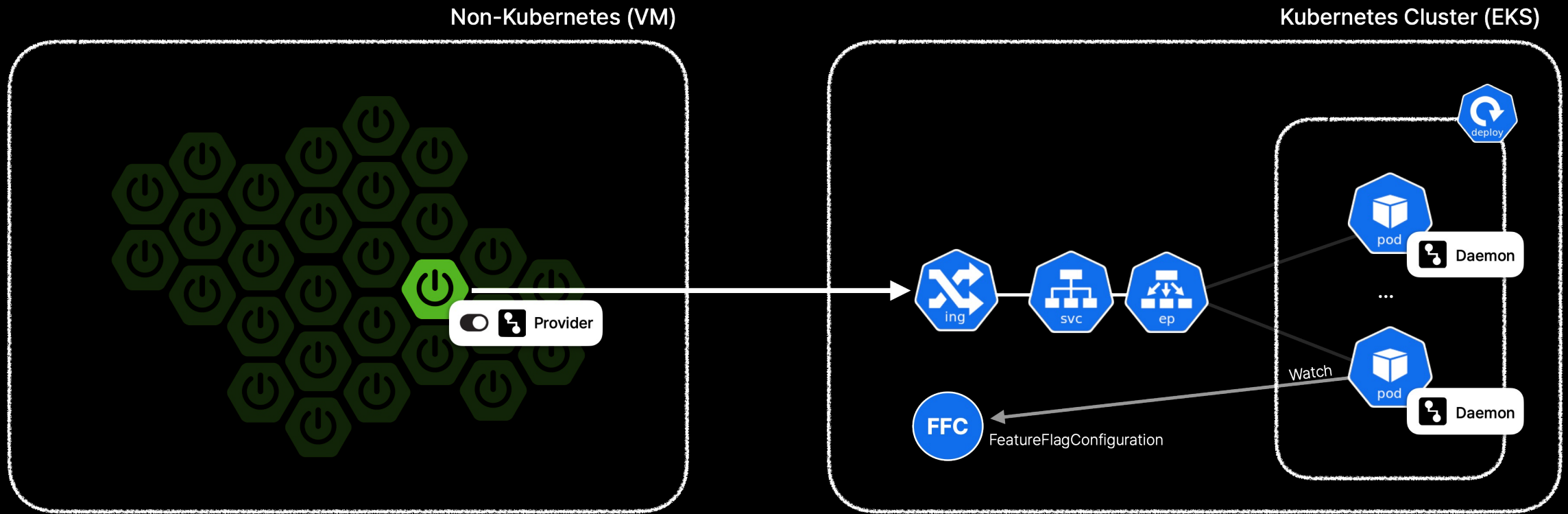


FFC FeatureFlagConfiguration

FFS FeatureFlagSource

VM+K8s (Deployment + Ingress)

11 TechTalk 2023



Flag Configuration

실시간 Flag 정의 수정 지원

다양한 타입의 Flag 정의 지원 Boolean, String, Number, JSON

상황에 따른 규칙 적용으로 특정 사용자 혹은 사용자 특성 타겟팅 지원

실험 용도의 Pseudo-Random 할당 지원

새로운 기능의 점진적 릴리즈 지원

여러 Flag Configuration의 Aggregation 지원

Default

```
flagd Flag Configuration - YAML (Default)
1 flags:
2   [flagKey]: _____ Flag 식별자
3   state: "ENABLED"
4   variants: _____ 값 - Alias 매칭
5     on: true
6     off: false
7   defaultVariant: "on" _____ Flag의 현재 Variant
8   targeting: {} _____ 조건 부여
```

```
Java Example
1 # 첫번째 파라미터 : Flag Key
2 # 두번째 파라미터 : 호출 실패되었을 때 기본 적용 값
3
4 if (client.getBooleanValue("[flagKey]", false)) {
5   return BetaFeature();
6 }
7 return LegacyFeature();
```

Targeting Rules

Context-Sensitive Evaluation

```
flagd Flag Configuration - YAML (Targeting Rules)
1 flags:
2   [flagKey]:
3     state: "ENABLED"
4     variants:
5       on: true
6       off: false
7     defaultVariant: "on"
8     targeting:
9       if:
10        - in:
11          - var:
12            - host
13              - 127.0.0.1
14
15
```

일치할 경우 Variant on
일치하지 않는 경우 off

JSONLogic 문법 적용

```
Java Example
1 # 첫째 파라미터 : Flag Key
2 # 둘째 파라미터 : 호출 실패되었을 때 기본 적용 값
3 # 셋째 파라미터 : EvaluationContext
4
5 Feature feature() {
6   if (client.getBooleanValue("[flagKey]", false, getContext())) {
7     return BetaFeature();
8   }
9   return LegacyFeature();
10 }
11
12 private EvaluationContext getContext() {
13   return new ImmutableContext(Map.of("host", "XXX.XXX.XXX.XXX"));
14 }
```

Targeting Rules

Fractional Operation

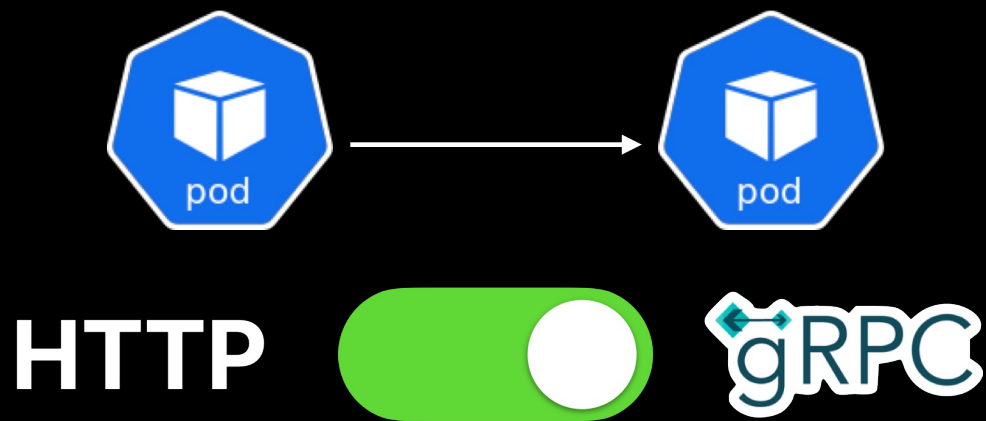
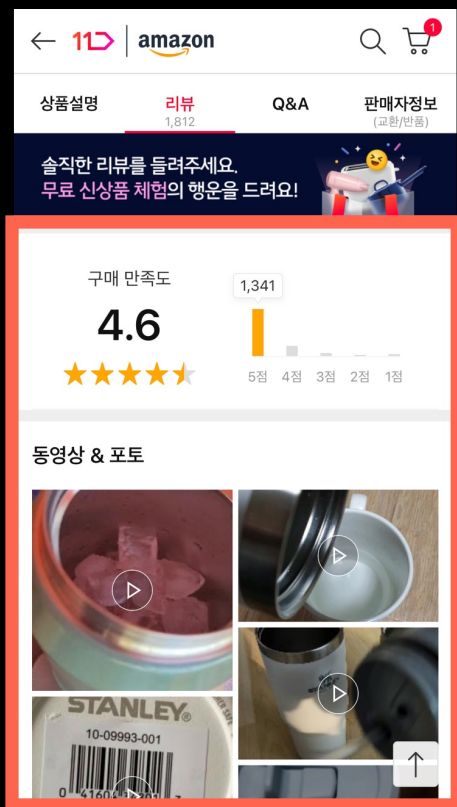
```
flagd Flag Configuration (Fractional)

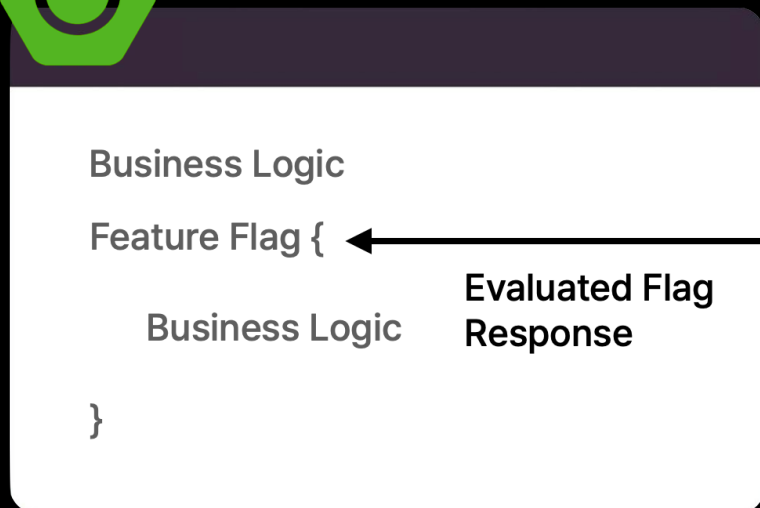
1 flags:
2   [flagKey]:
3     variants:
4       red: '#FF0000'
5       blue: '#0000FF'
6       green: '#00FF00'
7     defaultVariant: "red"
8     state: "ENABLED"
9     targeting:
10      fractional:
11        - var: email
12          - red
13            - 50
14          - blue
15            - 20
16          - green
17            - 30
```

Random하게 Split하는 기준이 되는 Context Key

EvaluationContext의 "email"을 기준으로 랜덤하게
50% → red / 20% → blue / 30% → green 할당

Lab





FeatureFlagConfiguration

Request



flagd

Evaluate Flag

```

1 dependencies {
2   implementation 'dev.openfeature:sdk:1.4.1'
3   implementation 'dev.openfeature.contrib.providers:flagd:0.6.0'
4 }
    
```




```

@Configuration
@EnableConfigurationProperties(FlagdProperties.class)
@RequiredArgsConstructor
@ConditionalOnProperty(value = "feature-flag.enabled", havingValue =
public class FeatureFlagConfiguration {

    private final FlagdProperties flagdProperties;

    @Bean
    public Client openFeatureClient(FeatureProvider provider) {
        final var instance = OpenFeatureAPI.getInstance();
        final var provider = new FlagdProvider(FlagdOptions.builder()
            .host(flagdProperties.host())
            .port(flagdProperties.port())
            .tls(flagdProperties.tls())
            .build());
        instance.setProvider(provider);
        return instance.getClient();
    }
}

```

별도 설정 없을 경우 localhost:8013

```

@Service
@RequiredArgsConstructor
class AmazonProductServiceImpl implements ProductService {
    private static final FLAG_IS_GRPC_ENABLED = "isGrpcEnabled";

    private final Client featureFlagClient;

    private final HTTPReviewService httpReviewService;
    private final GRPCReviewService grpcReviewService;

    /**
     * featureFlagClient.getBooleanValue()
     *   - 1st Parameter : FLAG KEY (FLAG명)
     *   - 2nd Parameter : Default Value (FLAG가 존재하지 않을 경우 반환 기본값)
     */
    public List<Review> getProductReviews(final long productId) {
        final boolean isGrpcEnabled = featureFlagClient.getBooleanValue(FLAG_IS_GRPC_ENABLED, false);

        if (isGrpcEnabled) {
            return grpcReviewService.getProductReviews(productId);
        }

        return httpReviewService.getProductReviews(productId);
    }
}

```

“isGrpcEnabled” Flag

defaultVariant: “OFF” -> “ON”

HTTP



```

2023-11-06 20:25:07.101 DEBUG [ ,614637d146f3720c,614637d146f3720c] 66132 --- [catalog-cache-1] e.r.s : Calling via HTTP
2023-11-06 20:25:08.099 DEBUG [ ,feeba5424bcd5470,feeba5424bcd5470] 66132 --- [catalog-cache-2] e.r.s : Calling via HTTP
2023-11-06 20:25:09.104 DEBUG [ ,c41f007bd60ae325,c41f007bd60ae325] 66132 --- [catalog-cache-2] e.r.s : Calling via HTTP
2023-11-06 20:25:10.131 DEBUG [ ,bb45e2e92f3fa1cf,bb45e2e92f3fa1cf] 66132 --- [catalog-cache-2] e.r.s : Calling via gRPC
2023-11-06 20:25:11.099 DEBUG [ ,dda96eee4a3b11c1,dda96eee4a3b11c1] 66132 --- [catalog-cache-2] e.r.s : Calling via gRPC
2023-11-06 20:25:12.096 DEBUG [ ,721df433a1b597ad,721df433a1b597ad] 66132 --- [catalog-cache-2] e.r.s : Calling via gRPC
2023-11-06 20:25:13.100 DEBUG [ ,bbb31f9e9bcb380b,bbb31f9e9bcb380b] 66132 --- [catalog-cache-1] e.r.s : Calling via gRPC
2023-11-06 20:25:14.093 DEBUG [ ,2fe12bdc0d7cf0c9,2fe12bdc0d7cf0c9] 66132 --- [catalog-cache-1] e.r.s : Calling via gRPC
2023-11-06 20:25:15.106 DEBUG [ ,b9233398097fb04d,b9233398097fb04d] 66132 --- [catalog-cache-1] e.r.s : Calling via gRPC
2023-11-06 20:25:16.102 DEBUG [ ,cc0fff6c46032278,cc0fff6c46032278] 66132 --- [catalog-cache-1] e.r.s : Calling via gRPC
    
```

Q&A

E-mail) jiwon_jeon@sk.com

Thank you