CLOUDZ LABS

Spinnaker를 활용한 배포 파이프라인 구성

2018.08.29

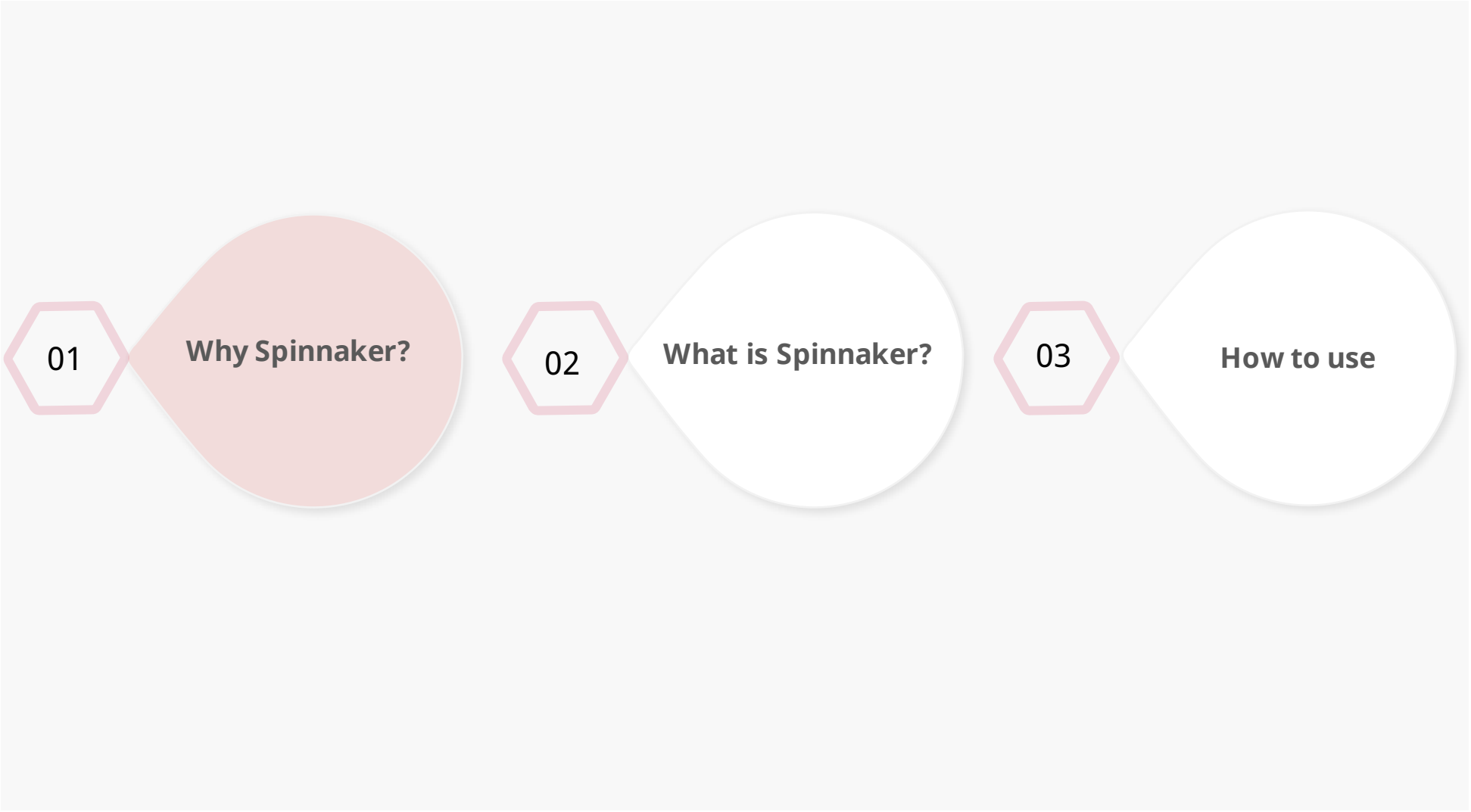
Copyright©2018 by DIGITAL LABS All rights reserved.

SK 주식회사

Table of Contents

Copyrightⓒ2018 by DIGITAL LABS All rights reserved.

Spinnaker

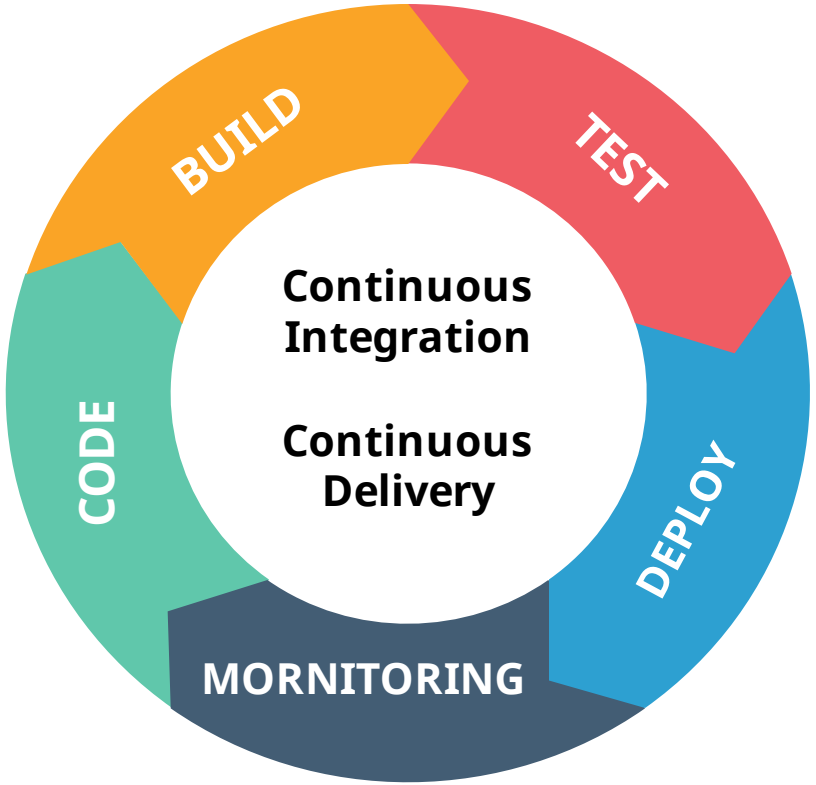


Copyrightⓒ2018 by DIGITAL LABS All rights reserved.

Why Spinnaker?

Continuous Integration, Continuous Delivery

빠르고 지속적으로 질 좋은 시스템을 제공하기 위해 자동화된 Build, Test, Deploy를 파이프라인으로 구성하여
애플리케이션의 ***지속적 통합***, ***지속적 배포***가 가능해야 함

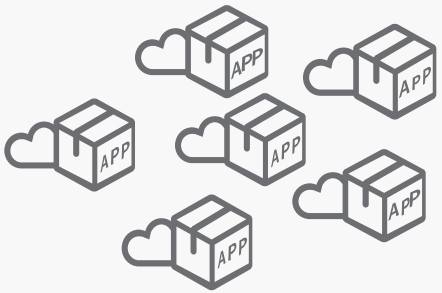


Copyright©2018 by DIGITAL LABS All rights reserved.

Why Spinnaker?

Monolitic

Microservice Architecture



Cloud Application

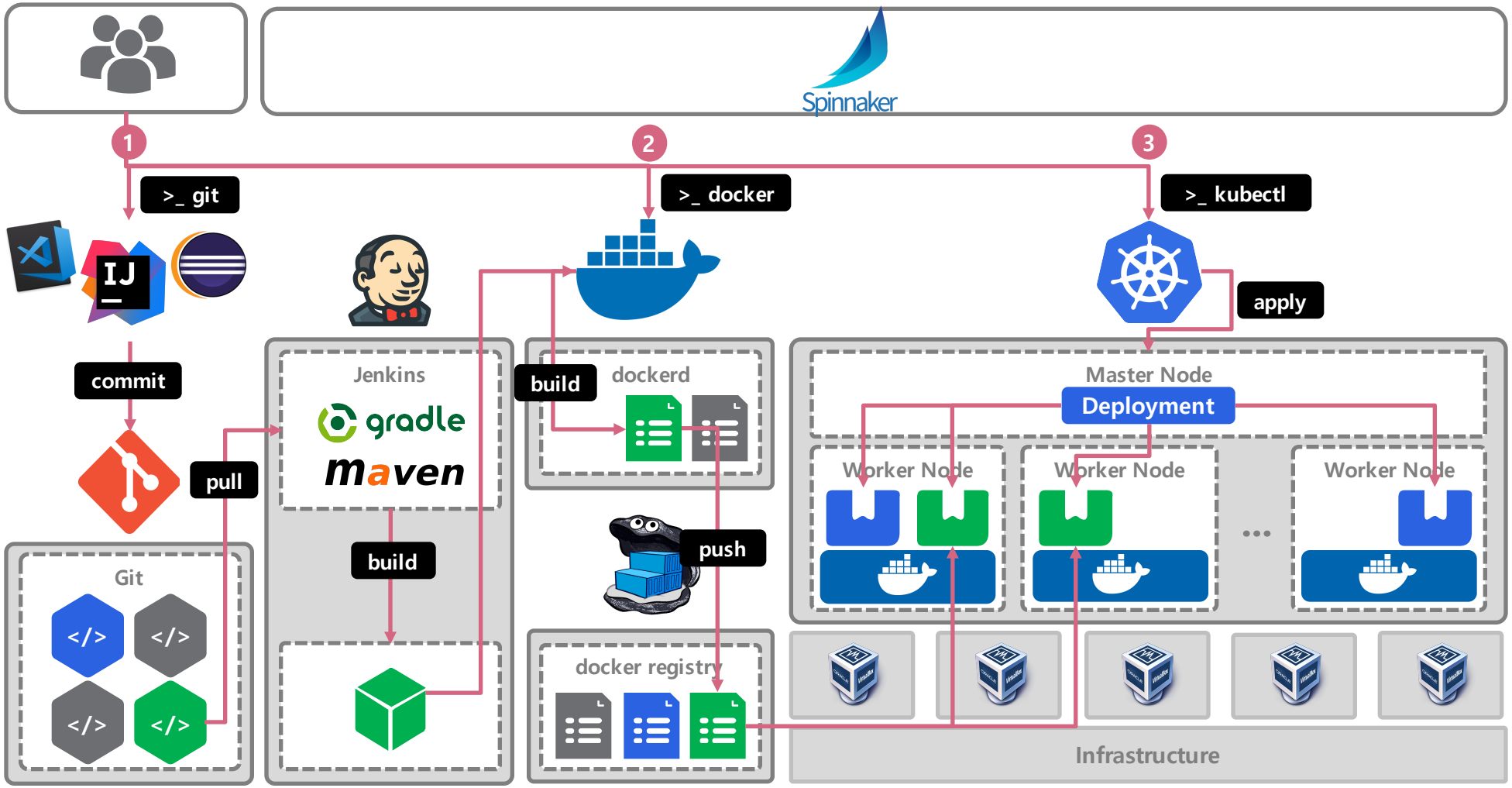
Mutable Infrastructure	Immutable Infrastructure
CHEF, ANSIBLE, puppet labs, CFEngine, SALTSTACK	?

Copyright©2018 by DIGITAL LABS All rights reserved.

Why Spinnaker?

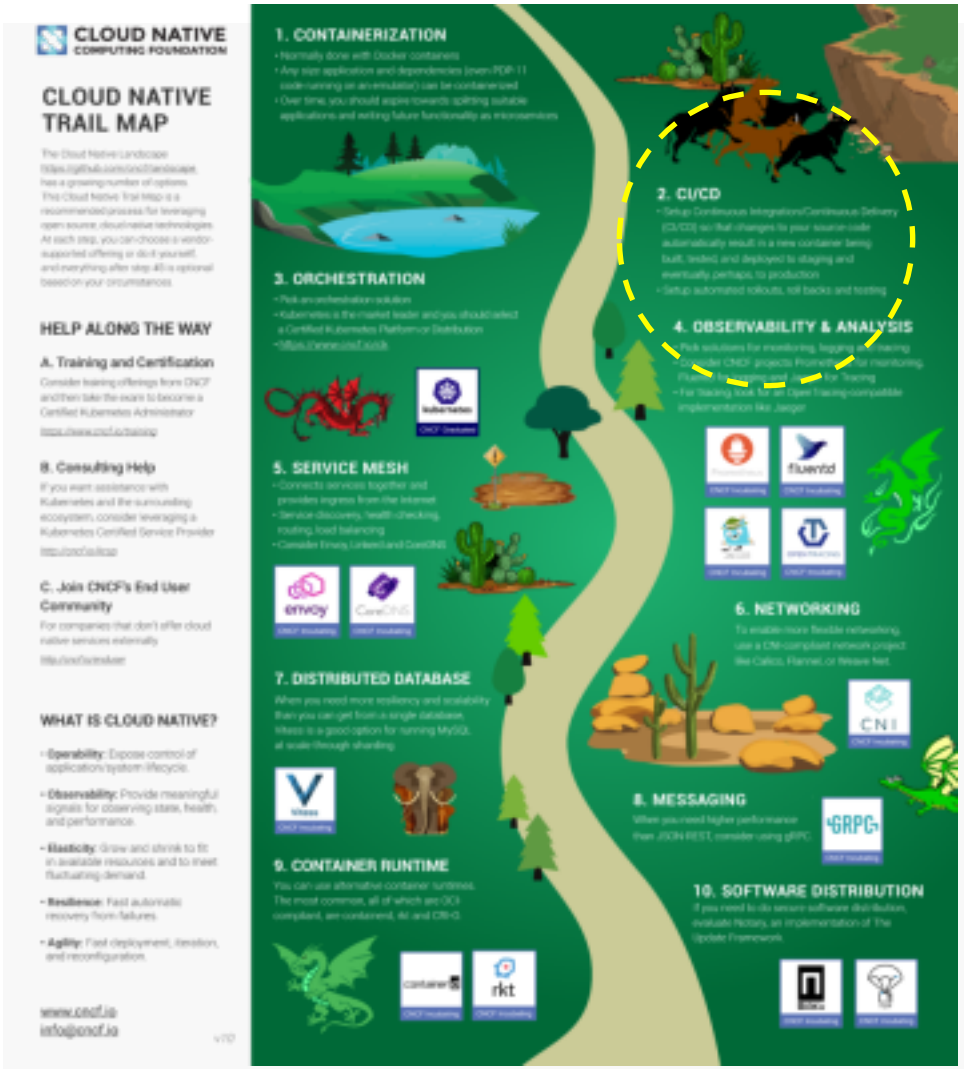
Copyright©2018 by DIGITAL LABS All rights reserved.

CI/CD usiing Spinnaker



Copyright©2018 by SK CLOUDZ LABS All rights reserved.

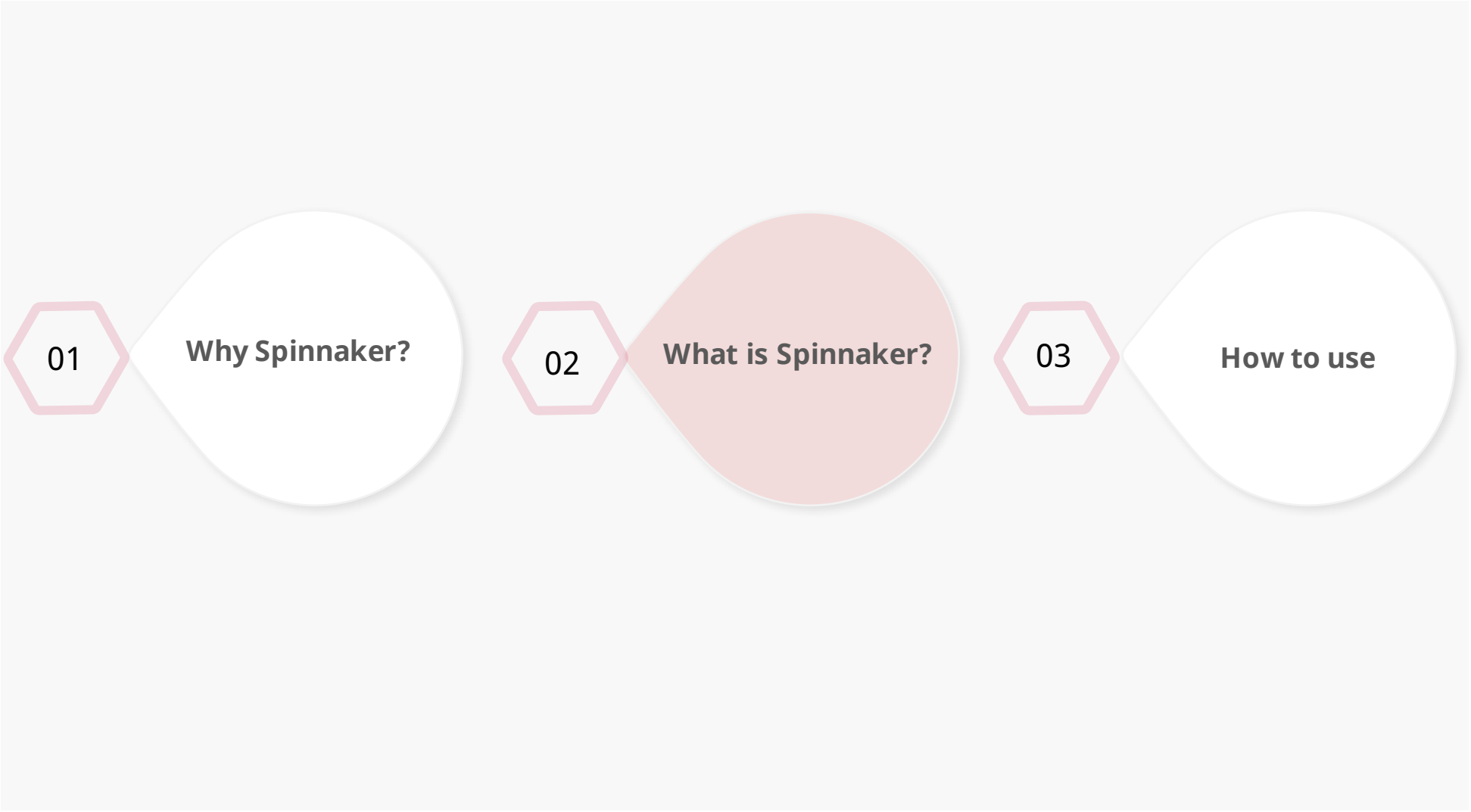
Why Spinnaker?



> *Setup Continuous Integrating/Continuous Delivery*
>
> *Setup automated rollouts, roll backs and testing*

Copyright©2018 by DIGITAL LABS All rights reserved.

Spinnaker

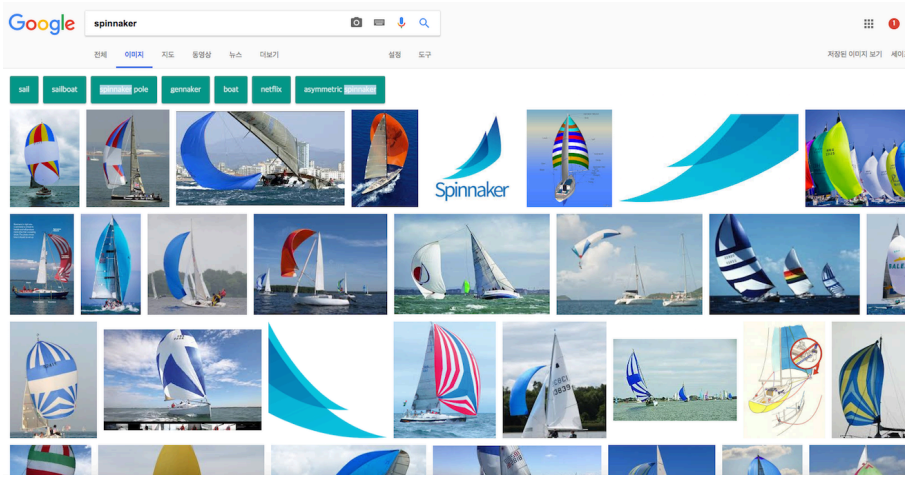


Copyrightⓒ2018 by DIGITAL LABS All rights reserved.

What is Spinnaker?

Spinnaker의 정의

영어 사전> 경주용 요트에 추가로 다는 큰 돛

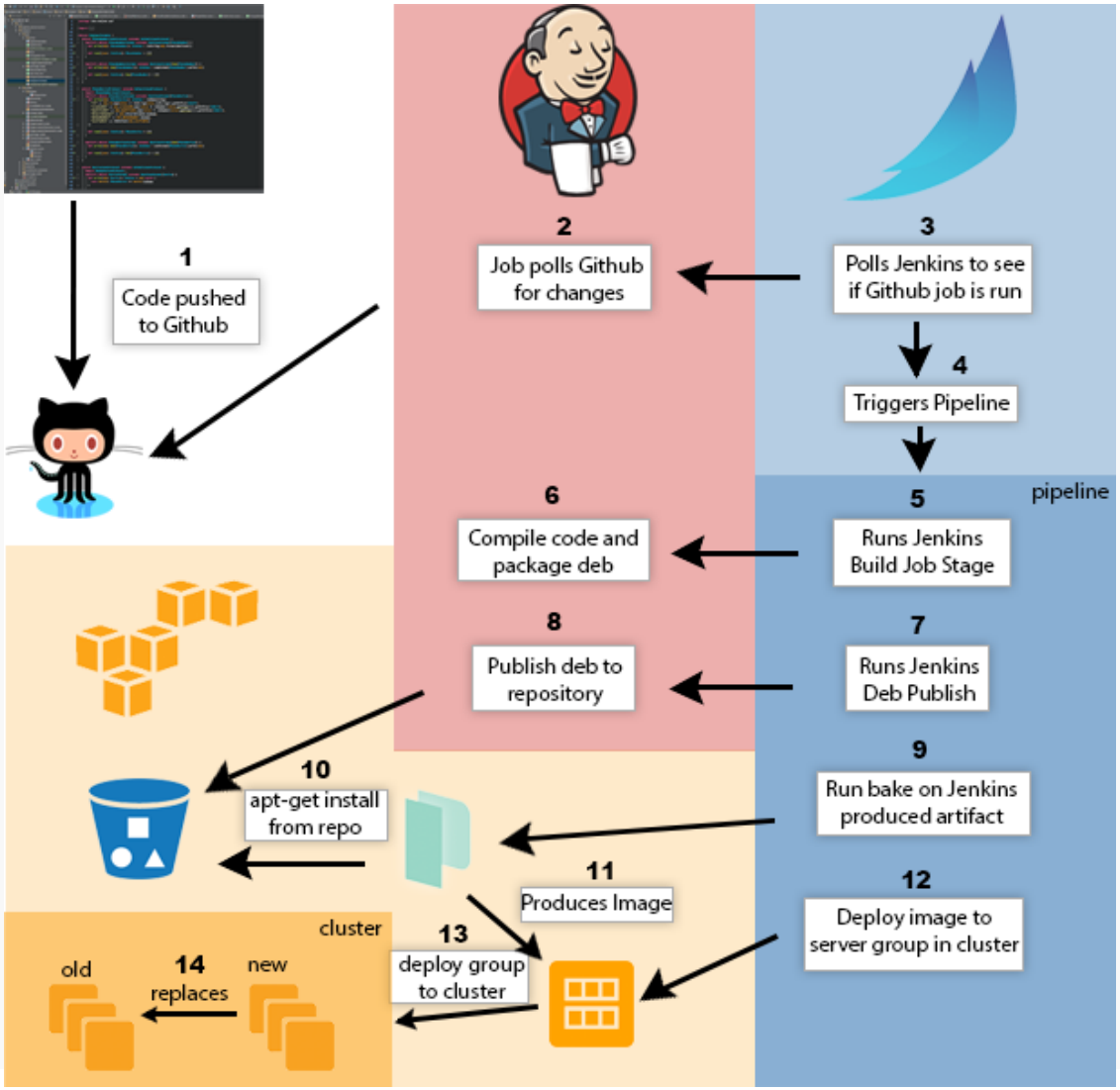


Spinnaker

continuous delivery를 위한
배포 자동화 관리 솔루션

Copyright©2018 by DIGITAL LABS All rights reserved.

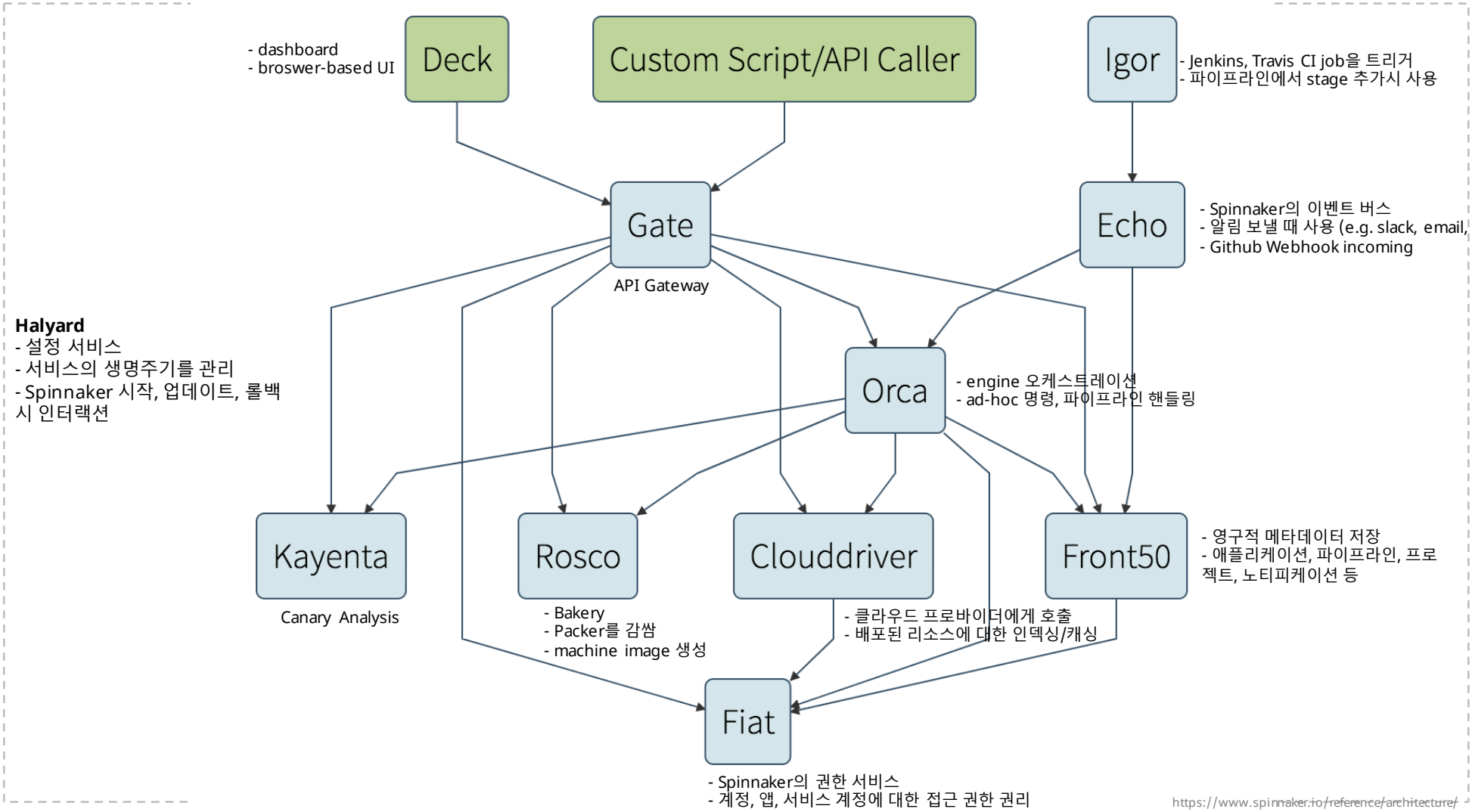
What is Spinnaker?



Copyright©2018 by DIGITAL LABS All rights reserved.

What is Spinnaker?

Spinnaker Architecture



https://www.spinnaker.io/reference/architecture/

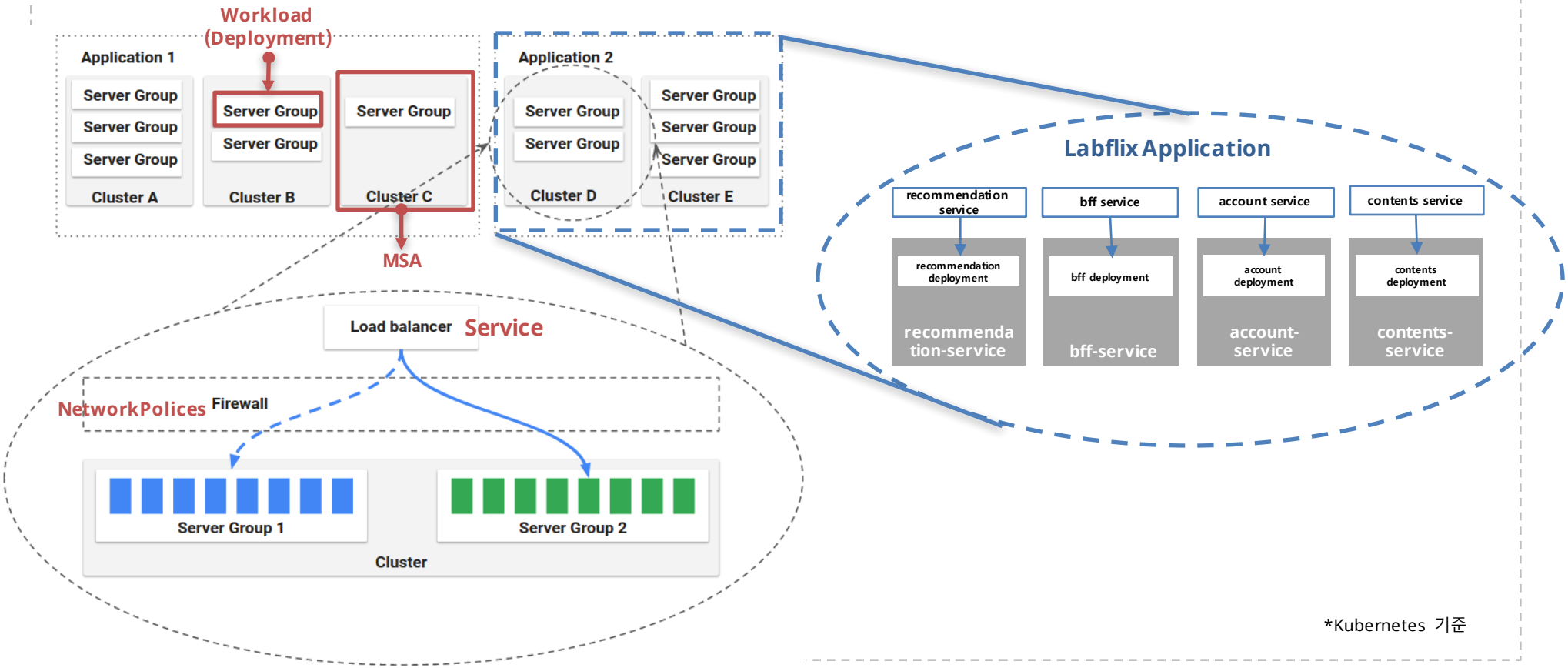
Copyright©2018 by DIGITAL LABS All rights reserved.

What is Spinnaker?

Spinnaker Concept – 1 Application Management

1 Application Management *클라우드에 있는 리소스를 조회, 관리*

Server Group < Cluster < Application



*Kubernetes 기준

Copyright©2018 by DIGITAL LABS All rights reserved.

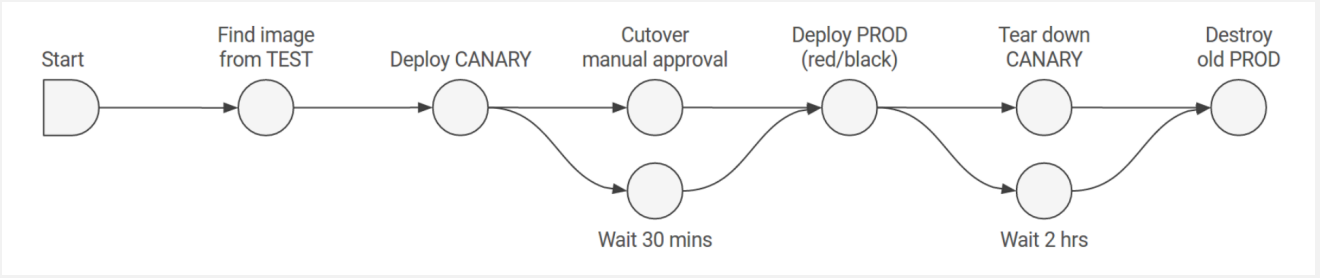
What is Spinnaker?

Spinnaker Concept – 2 Application Deployment

2 Application Deployment *지속적인 배포 워크 플로우 구성하여 배포를 자동화*

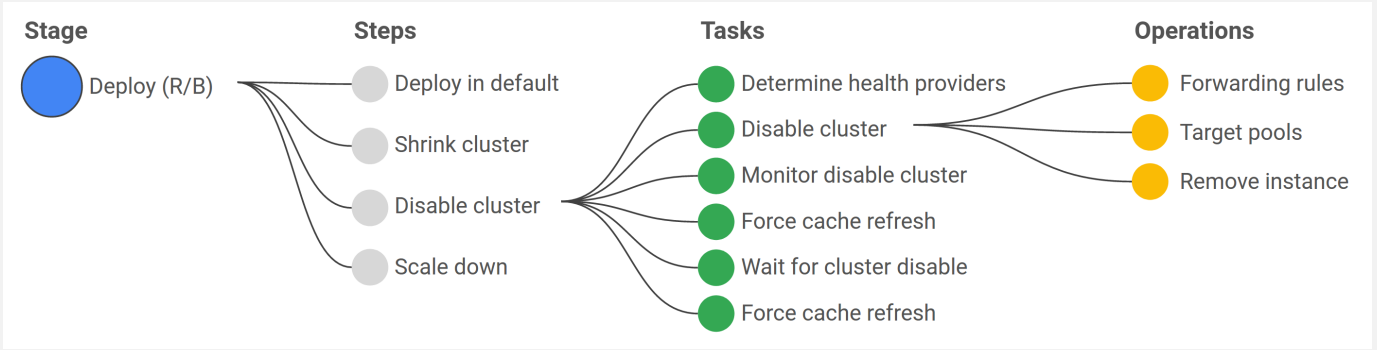
Pipeline

- Spinnaker의 배포 관리의 핵심
- 일련의 stage들로 구성
- Application을 생성
- 수동 또는 automated trigger로 시작



Stage

- Pipeline을 구성하는 가장 작은 단위
- 사용자가 수행하려는 동작을 설정

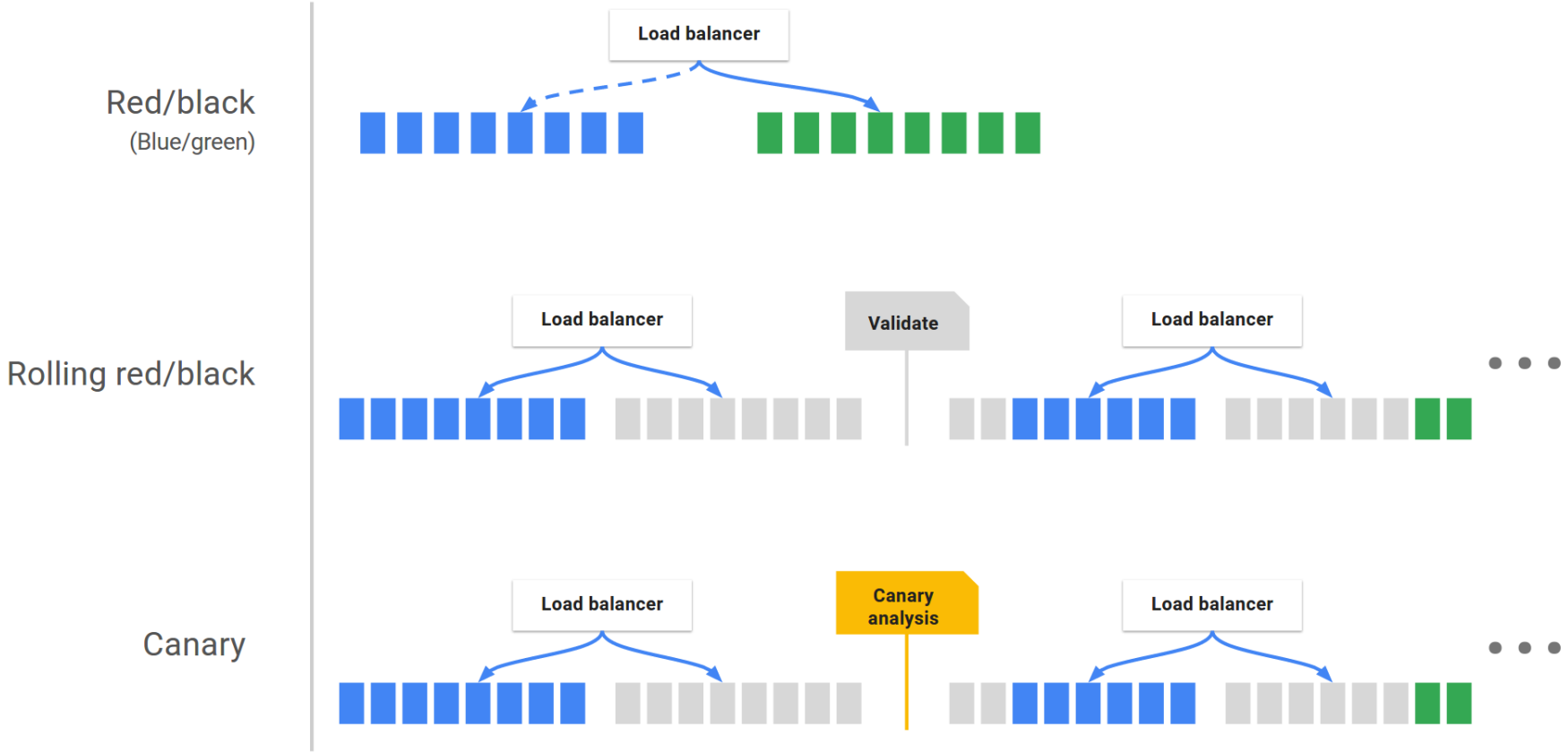


Copyright©2018 by DIGITAL LABS All rights reserved.

What is Spinnaker?

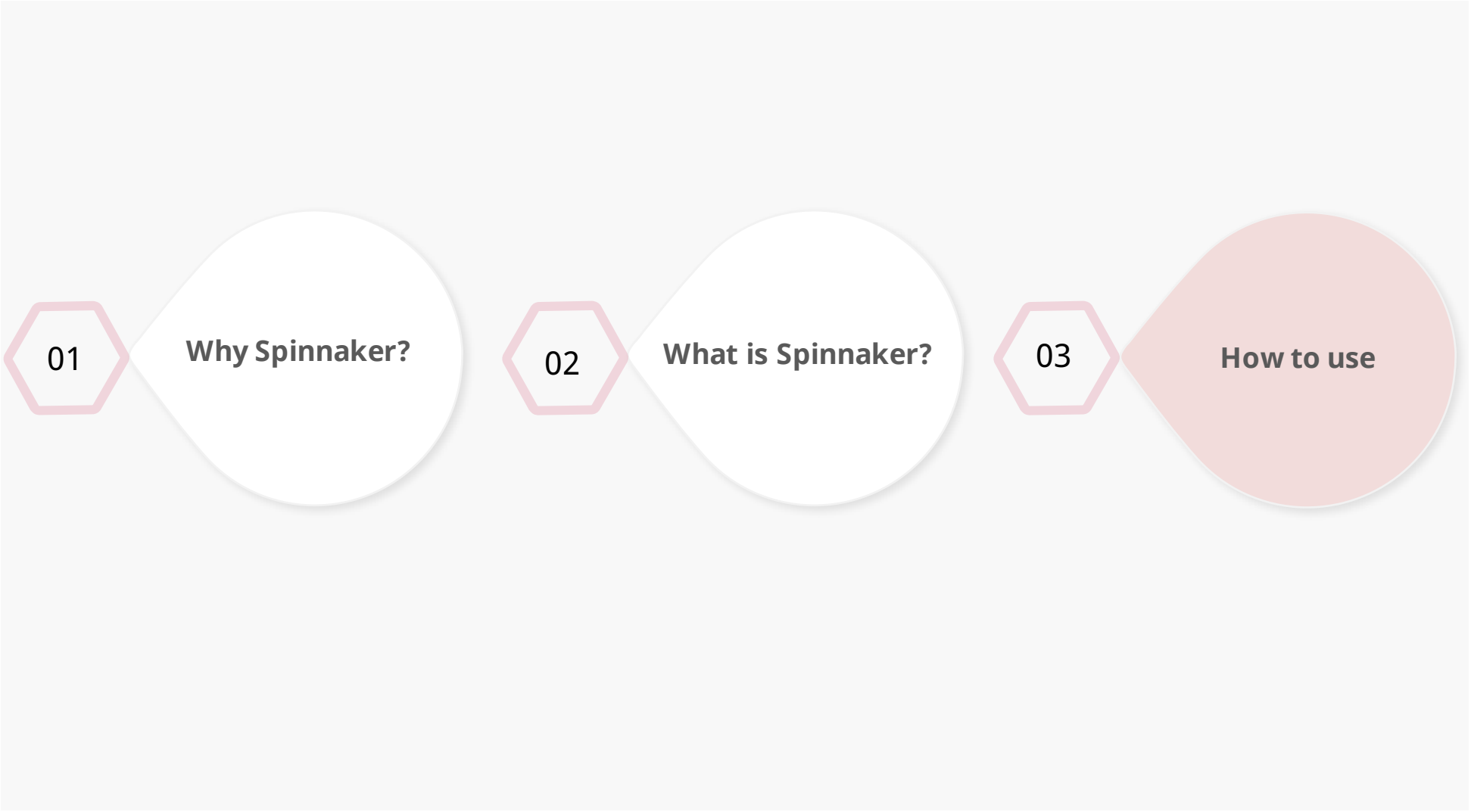
Spinnaker Concept – 2 Application Deployment

2 Application Deployment *다양한 배포 전략을 기본적으로 제공*



Copyright©2018 by DIGITAL LABS All rights reserved.

Spinnaker



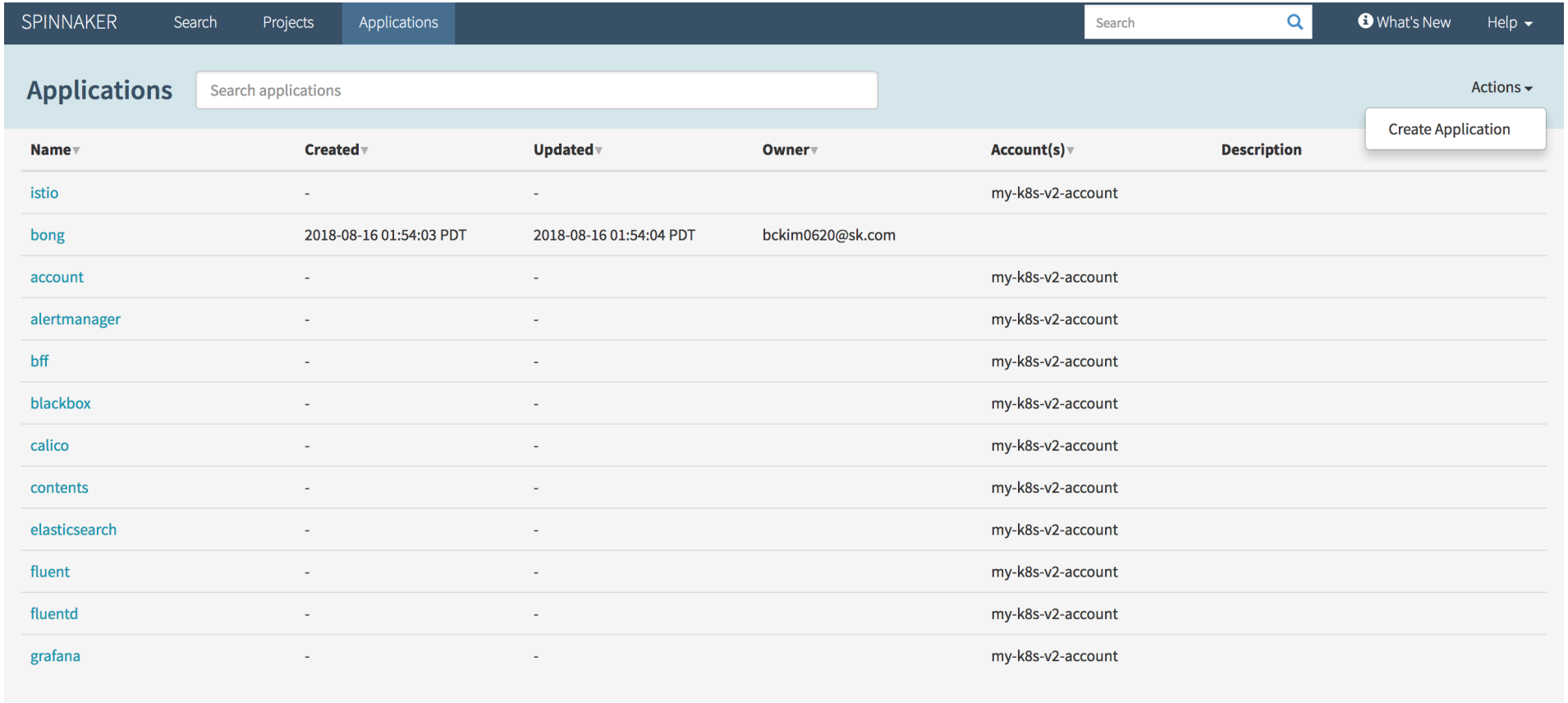
Copyrightⓒ2018 by DIGITAL LABS All rights reserved.

How to use

Spinnaker 구성요소 - Application

Application

- Spinnaker가 정의한 논리적 컨테이너 서비스의 집합
- **사용자가 직접 구성하여 사용**



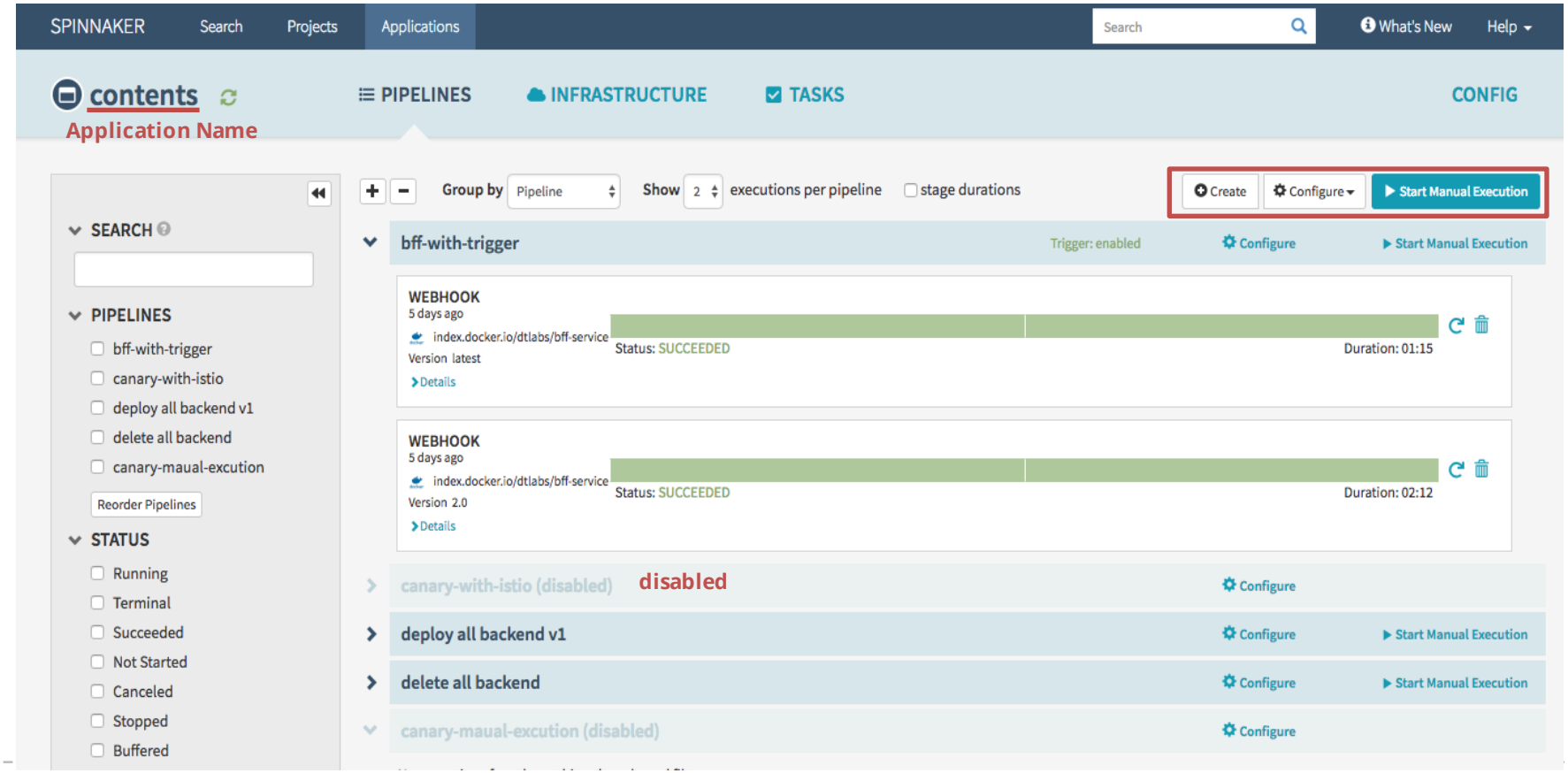
Copyright©2018 by DIGITAL LABS All rights reserved.

How to use

Spinnaker 구성요소 - Pipeline

Pipeline

- Pipeline목록, 실행 과정, 결과 및 자세한 정보를 볼 수 있음
- Pipeline 생성, 설정, 수동 실행



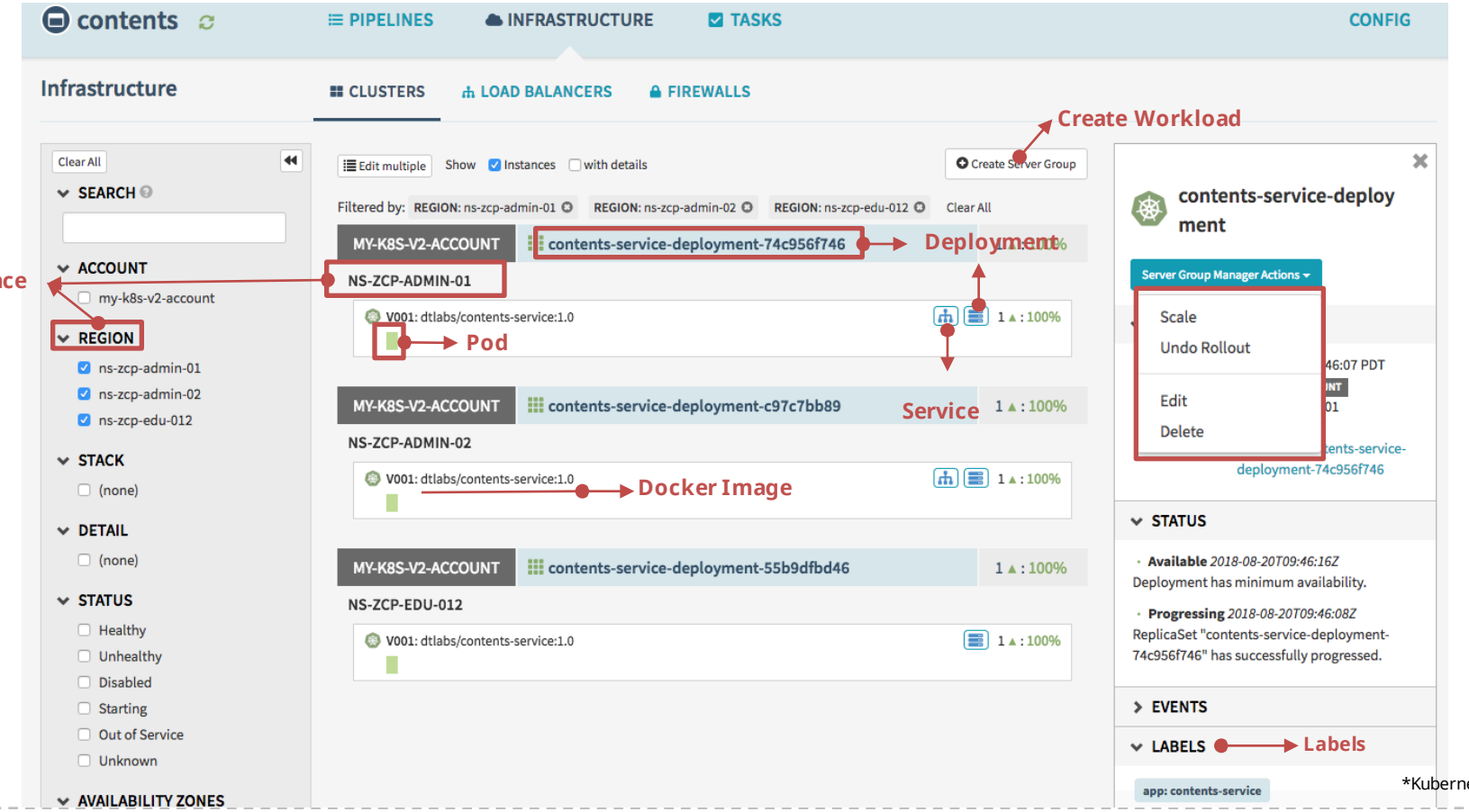
Copyright©2018 by DIGITAL LABS All rights reserved.

How to use

Spinnaker 구성요소 - Infrastructure

Cluster

- Application에 대한 인프라 정보를 조회할 수 있음
- Multi Cloud Provide의 Global한 배포 관리
- 실행 환경/배포 상태, 개별 인스턴스에 대한 메타데이터 조회
- Kubernetes의 Workload 정보 확인 및 설정 변경, 롤아웃 등을 UI로 수행할 수 있음



*Kubernetes 기준

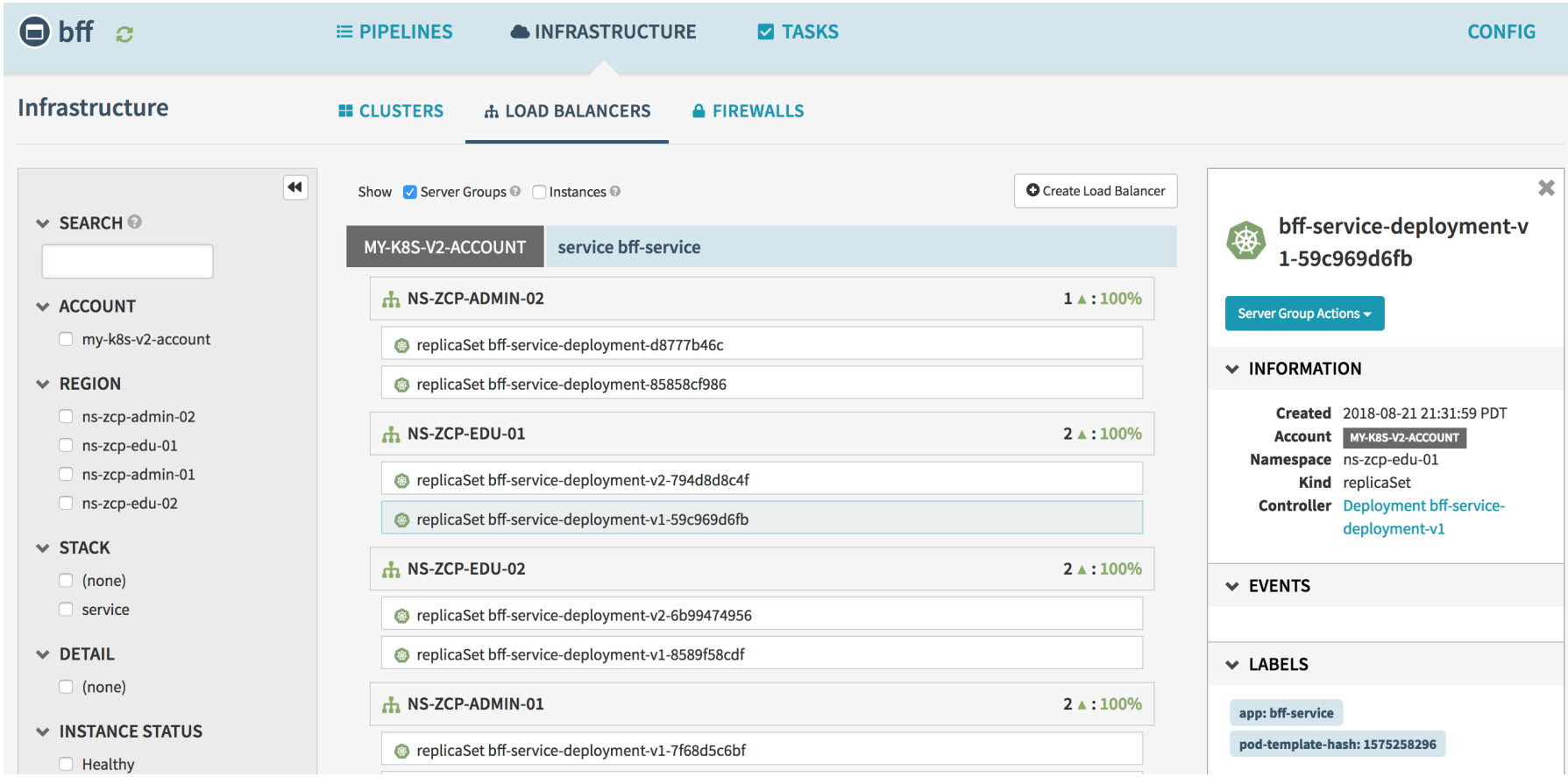
Copyright©2018 by DIGITAL LABS All rights reserved.

How to use

Spinnaker 구성요소 - Infrastructure

Load Balancers Kubernetes의 Service Controller

Firewall Kubernetes의 NetworkPolicy

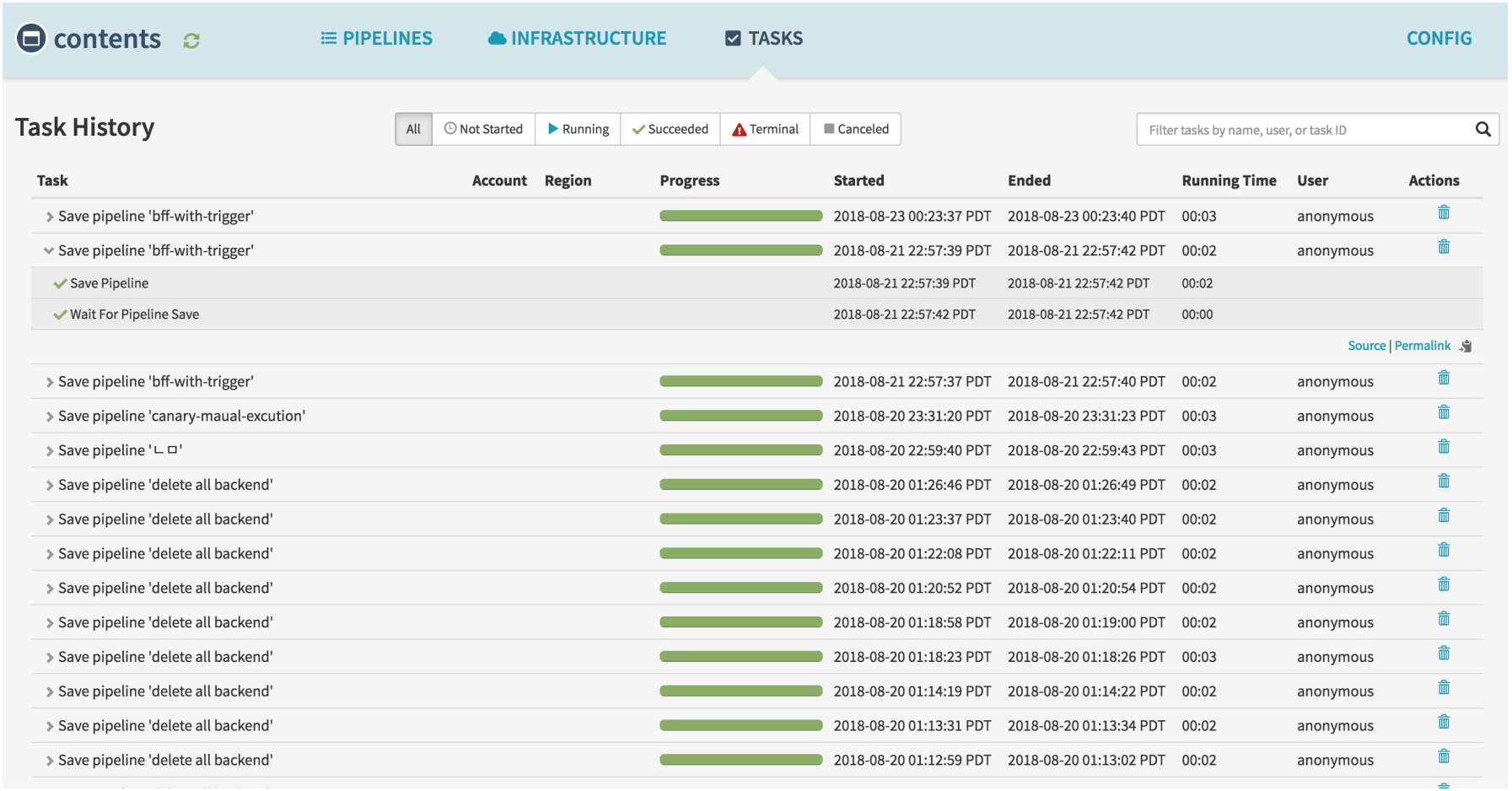


Copyright©2018 by DIGITAL LABS All rights reserved.

How to use

Spinnaker 구성요소 - Task

Task 해당 Application에서 동작하고 있는 task를 한눈에 관리할 수 있는 뷰



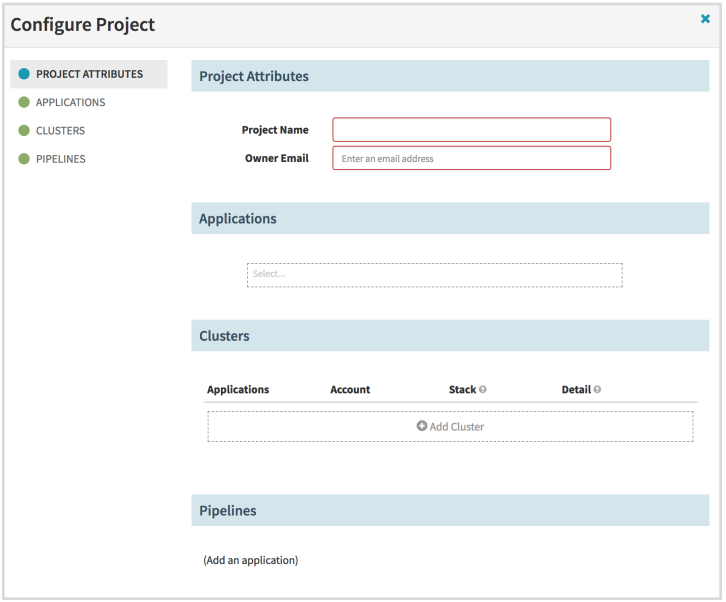
Copyright©2018 by DIGITAL LABS All rights reserved.

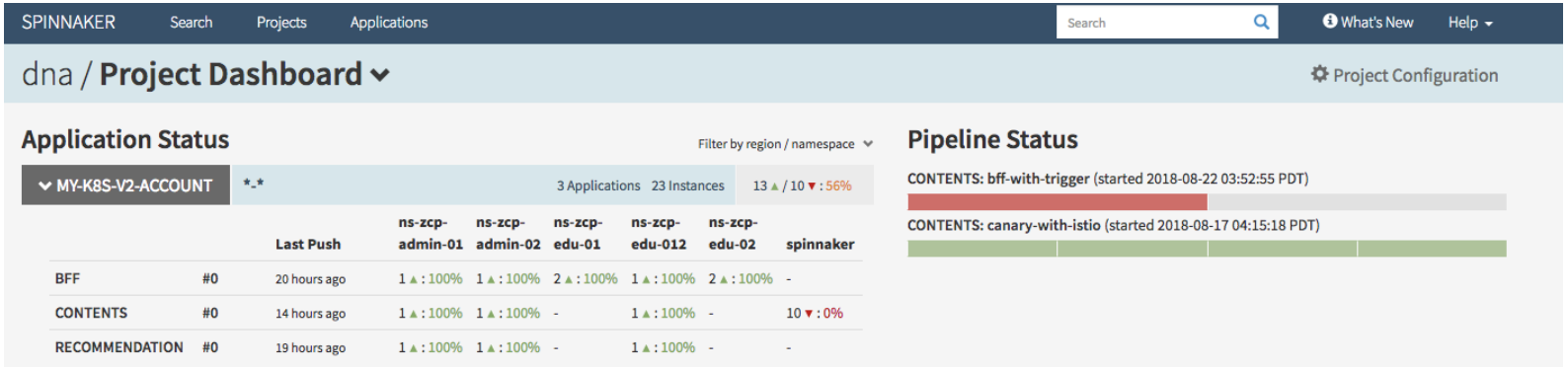
How to use

Spinnaker 구성요소 – Project

Project

사용자가 관리하려는 Cluster, Application, Pipeline의 집합





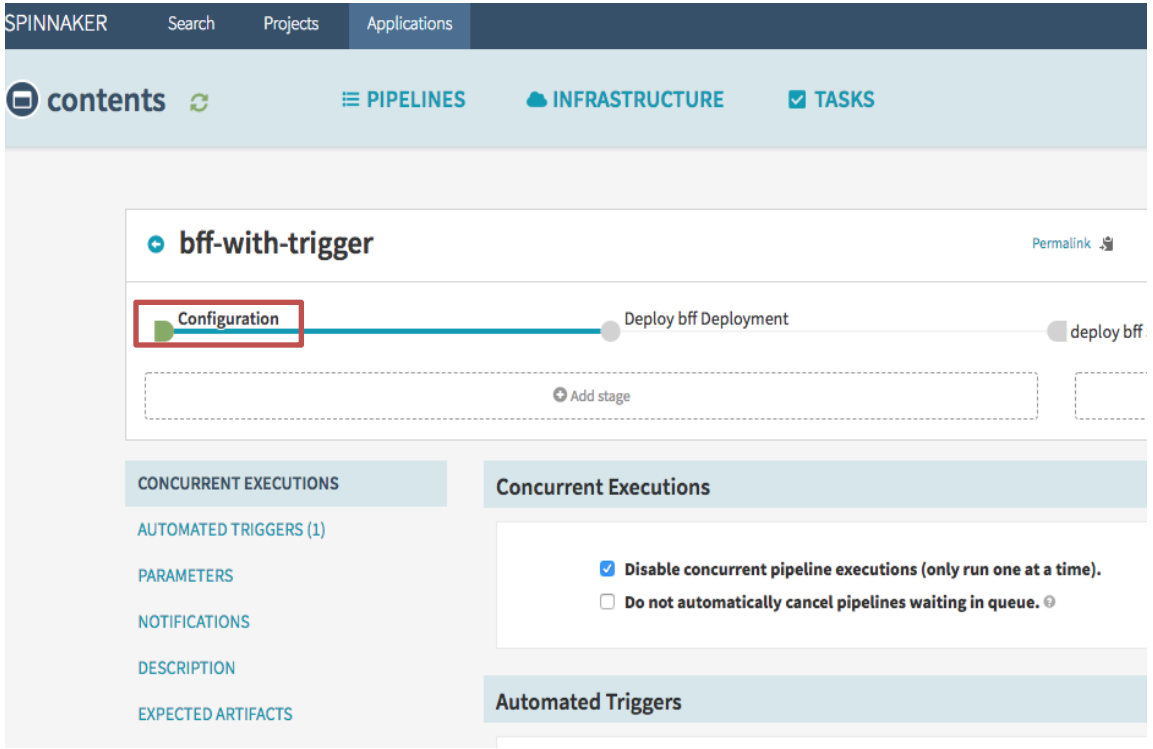
Copyrightⓒ2018 by DIGITAL LABS All rights reserved.

How to use

Pipeline 생성

1. Configuration 설정

parameter, artifact, 알림 등 Pipeline 수행에 필요한 정보를 설정하는 필수 단계



Bff-with-trigger pipieline 설정 화면

설정 항목

AUTOMATED TRIGGERS 트리거링 설정

PARAMETERS 사용자로부터 입력 받을 파라미터 정보 설정

NOTIFICATIONS 파이프라인 시작, 완료, 실패 시 수신을 여부 설정함. email, Slack 등 지원

DESCRIPTION 파이프라인에 대해 적고 싶은 내용을 적는 곳

EXPECTED ARTIFACTS 해당 파이프라인이 실행될 때 산출될 것으로 기대되는 아티팩트 정의

Copyright©2018 by DIGITAL LABS All rights reserved.

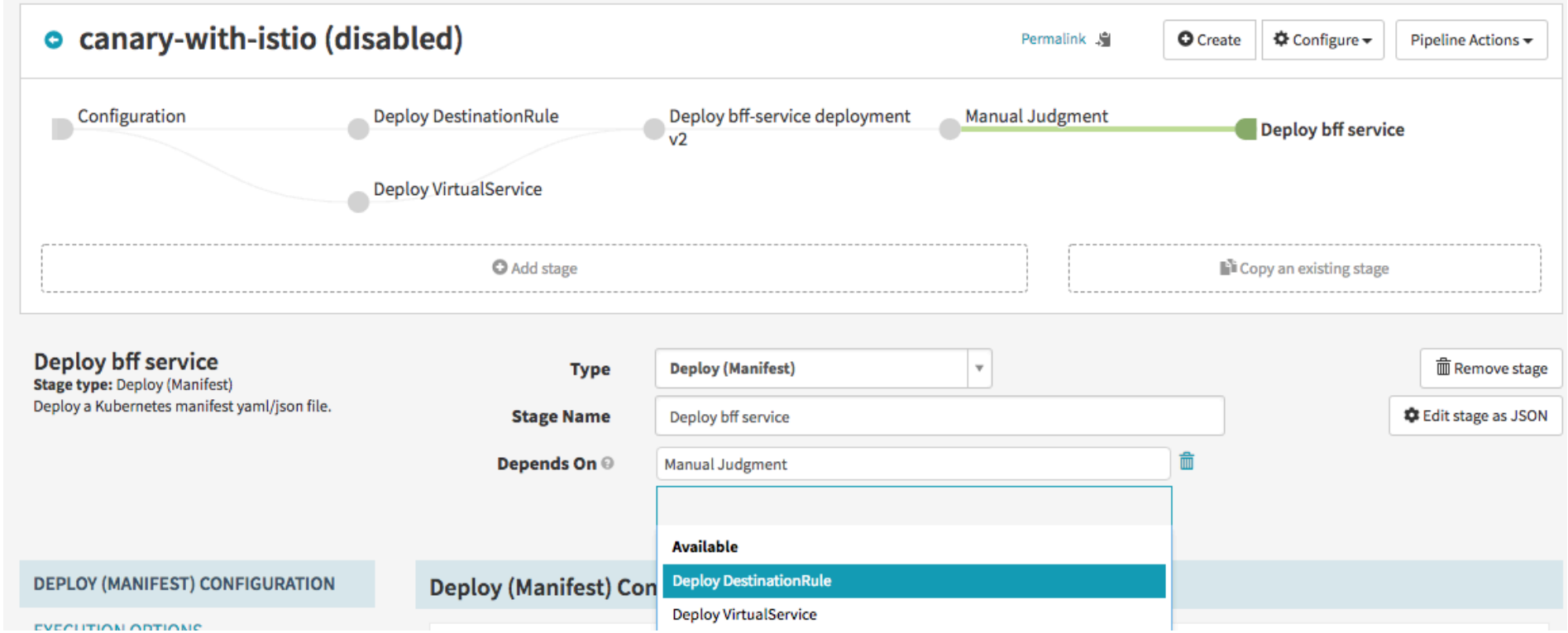
How to use

Pipeline 생성

2. Stage 설정

Stage 기본 설정

- Stage Type: 다른 stage copy 가능
- Pipeline명
- Stage 순서: Depends On 속성으로 설정



Copyright©2018 by DIGITAL LABS All rights reserved.

How to use

Pipeline 생성

Stage Type (K8S V2 기준)

Bake Image Baking. Helm 사용

Check Preconditions 클러스터 사이즈나 정규식을 이용한 상태 점검

Delete K8S 객체 삭제

Deploy 배포

Find Artifact from Execution 다른 파이프라인을 실행한 결과물 바인드

Find Artifact from Resource K8S 리소스를 찾아서 바인드

Manual Judgement 사용자의 승인

Jenkins Jenkins job 실행

Patch K8S의 객체를 Patch

Pipeline 지정한 파이프라인 실행

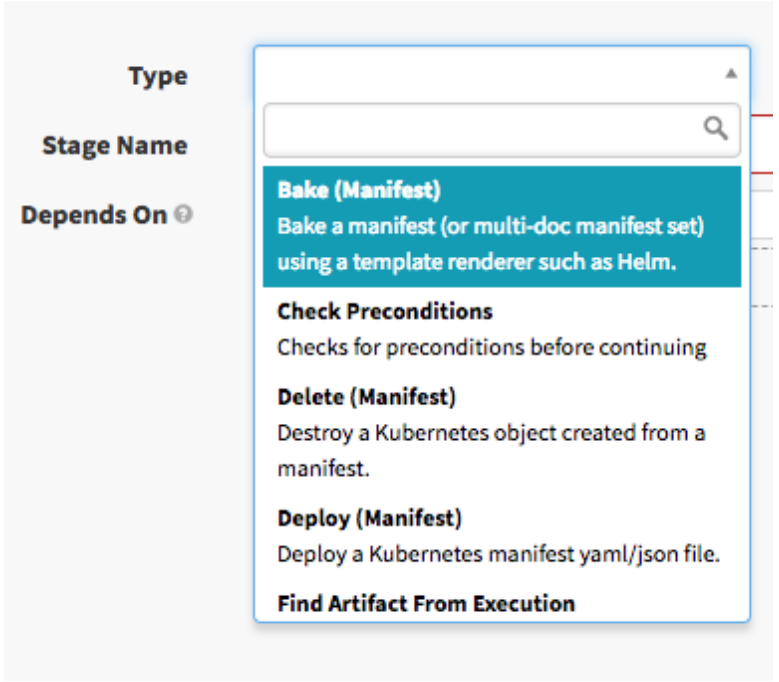
Scale Configuration K8S에 객체를 scaling

Script 스크립트 실행

Undo Rollout K8S 객체의 상태를 원하는 revision number로 롤아웃

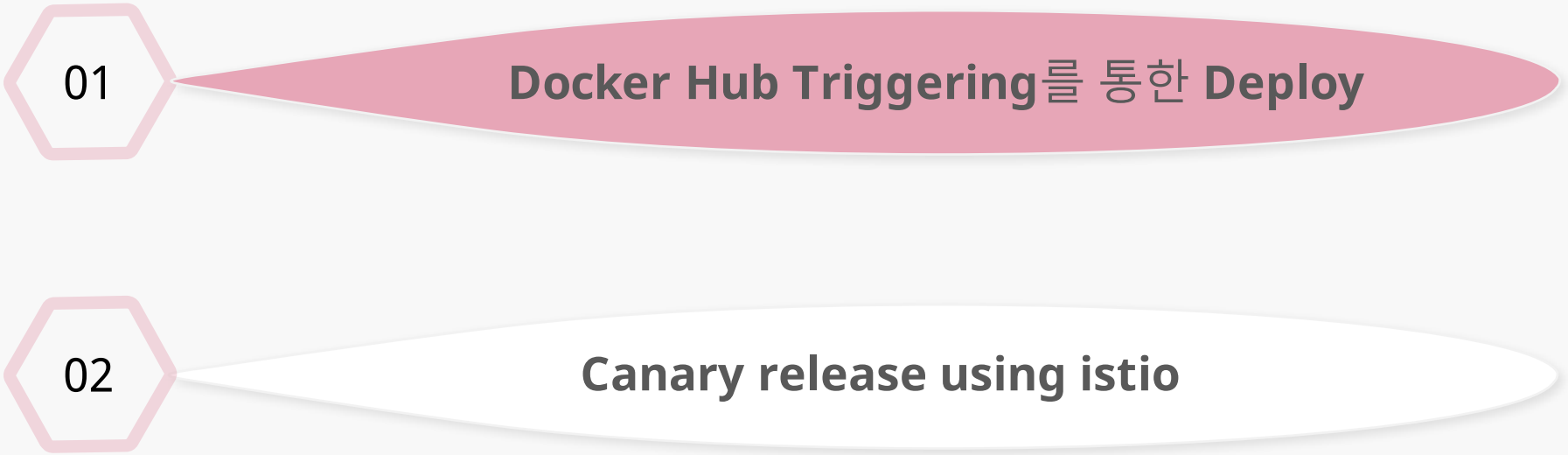
Wait 일정 시간 대기

Webhook HTTP의 GET, HEAD, POST, PUT, DELETE를 지정하여 Webhook 수행



Copyright©2018 by DIGITAL LABS All rights reserved.

Hands-on: Deploy Pipeline 생성하기

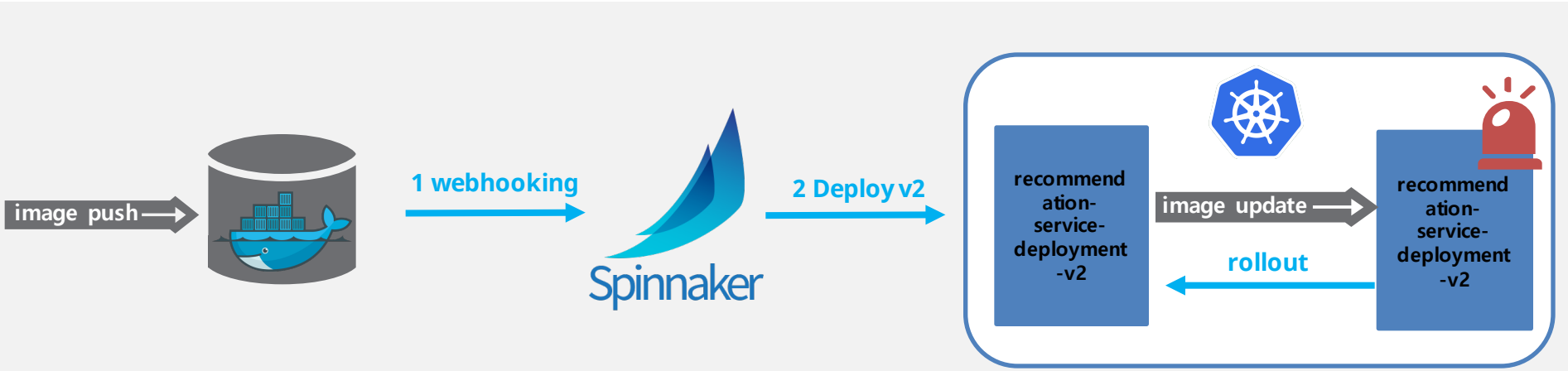


Copyrightⓒ2018 by DIGITAL LABS All rights reserved.

Docker hub triggering

일반 배포 자동화

시나리오



Docker image push로 인한 triggering을 시작으로 자동화 배포하고 잘못된 업데이트를 rollout

Copyright©2018 by DIGITAL LABS All rights reserved.

Docker hub triggering

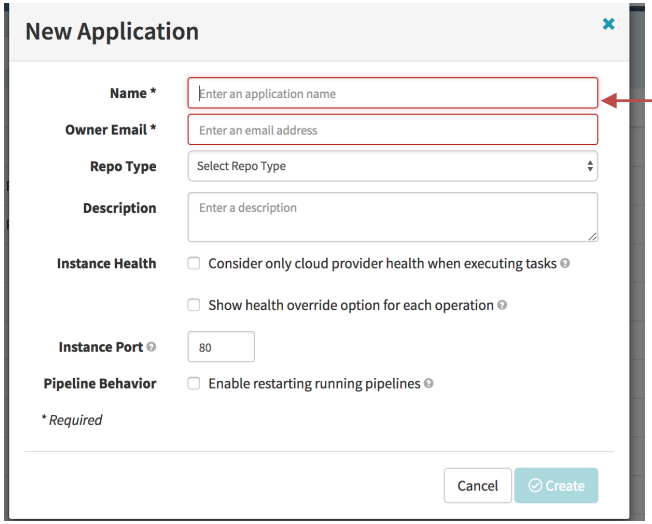
1,2 Application 생성

1 **Spinnaker Dashboard에 접속** http://spinnaker.zcp-dtlabs.jp-tok.containers.mybluemix.net/

2 **Application 생성**
2-1 상단의 ***Applications*** 클릭 > 우측 ***Actions*** 버튼을 클릭하여 ***Create Application*** 선택

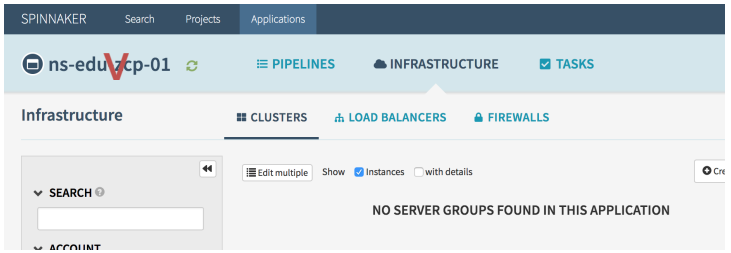
2-2 **Application 정보 입력**

- Application명과 Email 주소 입력
- ***Create*** 버튼 클릭



Labsflix${number}

✓ 생성된 Application 확인

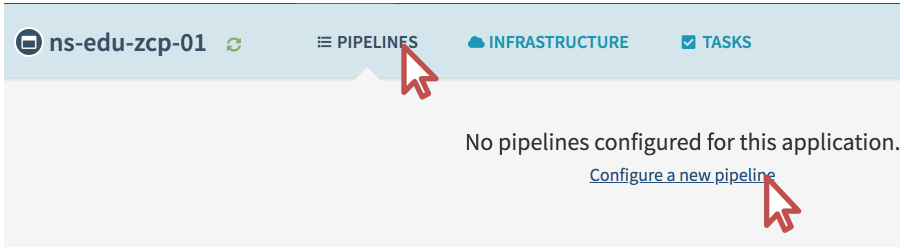


Copyrightⓒ2018 by DIGITAL LABS All rights reserved.

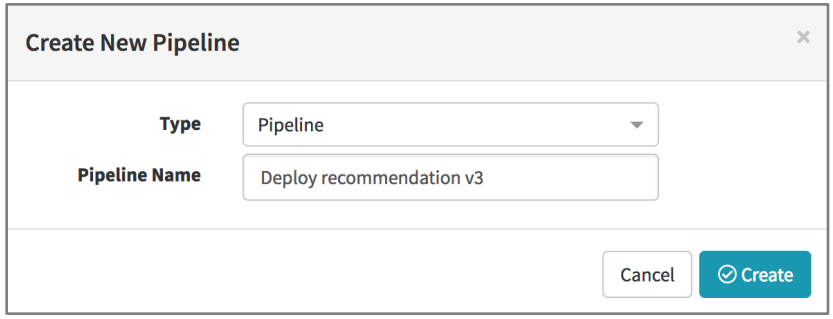
Docker hub triggering

3 Pipeline 생성

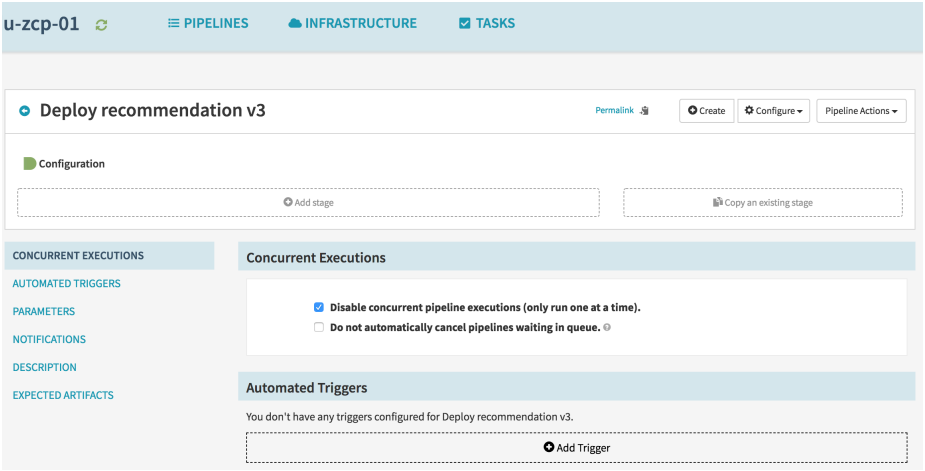
3-1 상단의 ***PIPELINES*** 클릭 – 하단의 Configure a new pipeline



3-2 ***Type: Pipeline*** 선택
Pipeline Name: 'Deploy recommendation v2' 입력
Create 버튼 클릭



✓ Pipeline 생성 확인



Copyright©2018 by DIGITAL LABS All rights reserved.

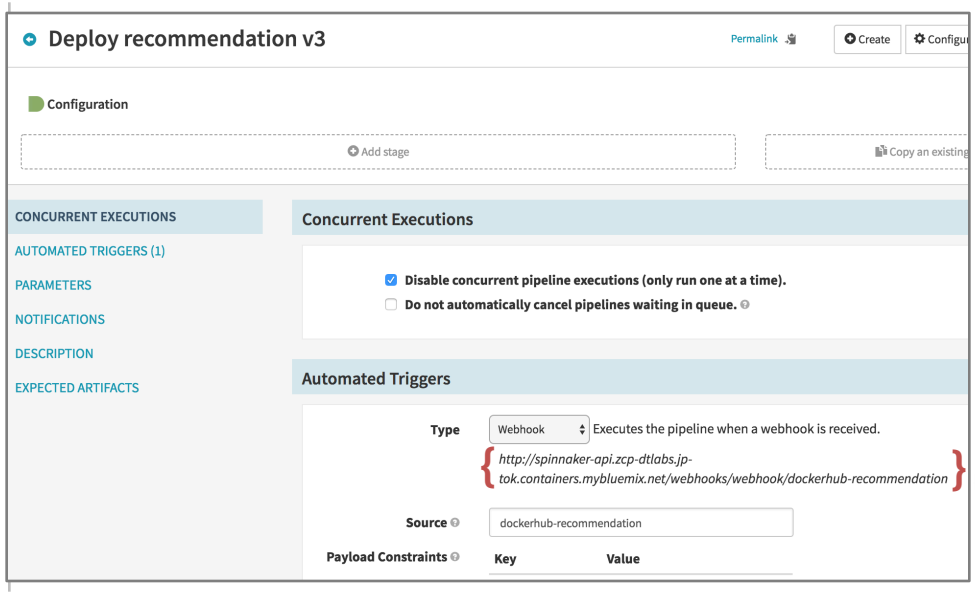
Docker hub triggering

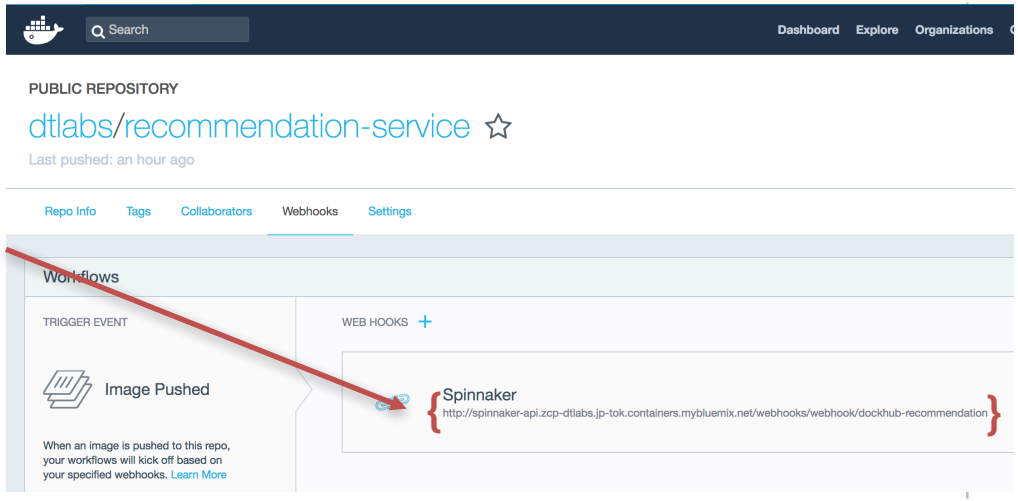
4 Docker Image Webhook 적용

Docker Image hooking에 사용할 특정 URL을 만든다.

Configuration stages의 'Automated Triggers'에서 다음과 같이 설정

- ***Type: Webhook*** 선택
- ***Source:*** domain 뒤에 올 URL name 입력
  dockerhub-recommendation 입력





Copyright©2018 by DIGITAL LABS All rights reserved.

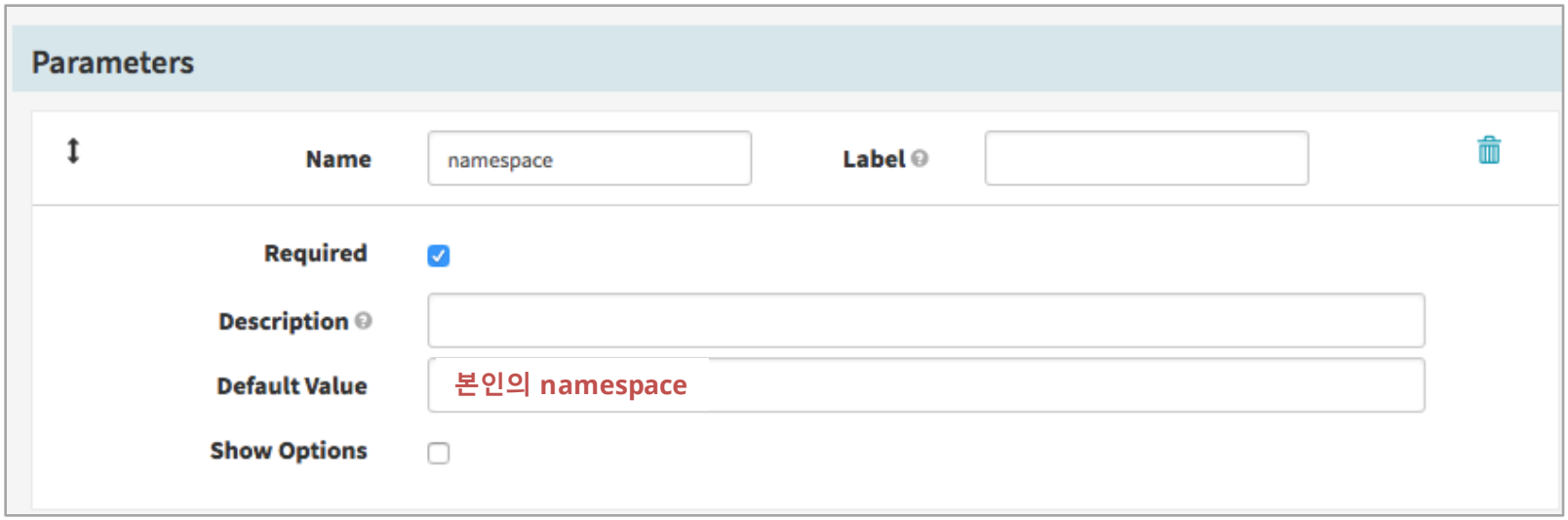
Docker hub triggering

4 Parameter 설정

Namespace용 parameter를 생성한다.

Configuration stages의 **Parameters**에서 다음과 같이 설정

- ***Name:*** namespace
- ***Default Value:*** 배포하려는 namespace



Copyright©2018 by DIGITAL LABS All rights reserved.

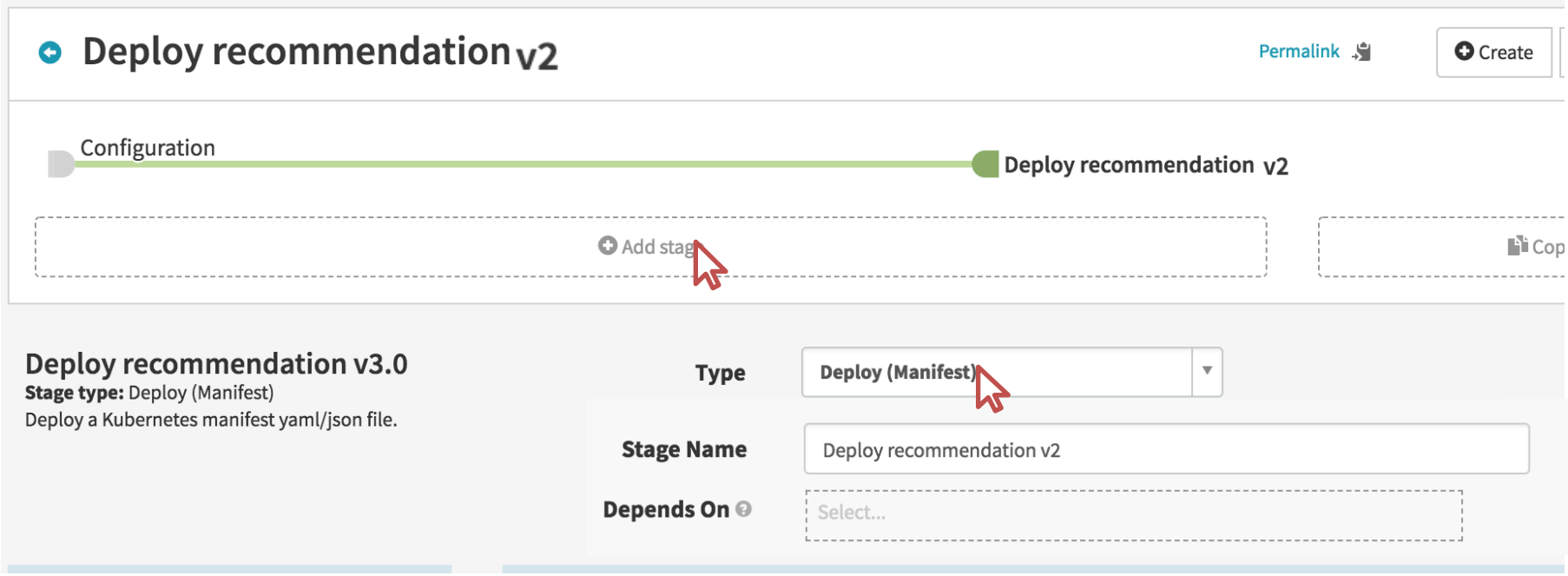
Docker hub triggering

5 Kubernetes에 Deployment 배포

배포를 위한 파이프라인을 생성한다.

5-1 ***Add stage*** 클릭 - 다음과 같이 설정

- ***Type: Deploy (Manifest)*** 선택
- ***Stage Name:*** Stage명 입력
  Deploy recommendation v2 입력



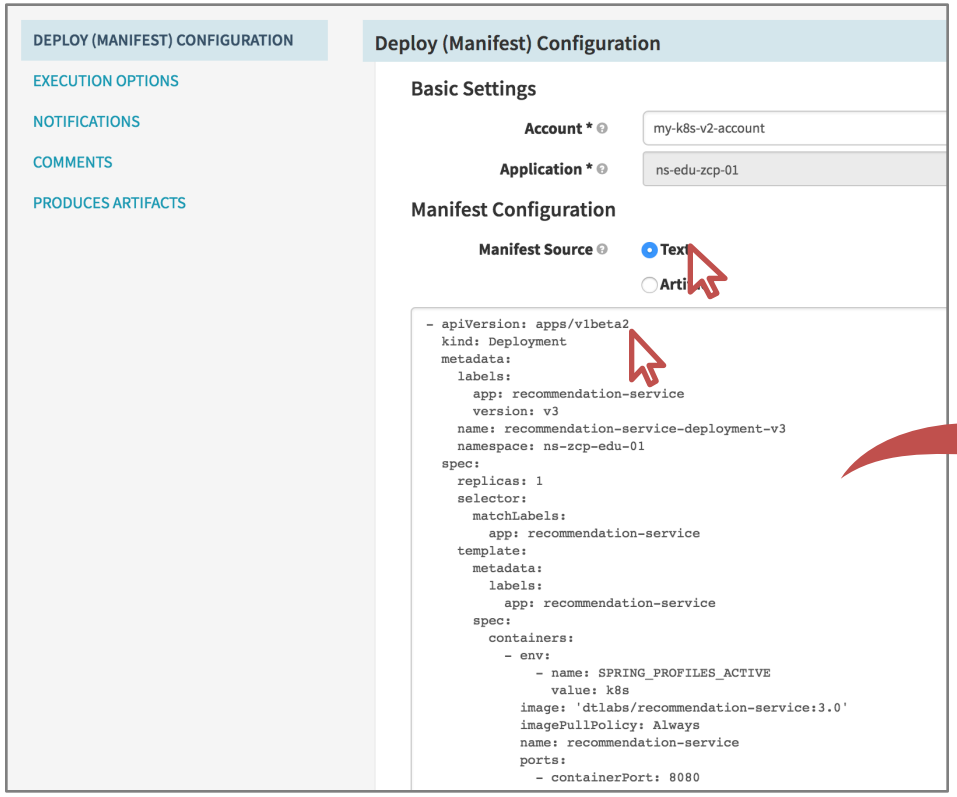
Copyright©2018 by DIGITAL LABS All rights reserved.

Docker hub triggering

5 Kubernetes에 Deployment 배포

배포할 Deployment의 매니페스트를 입력한다.

5-2 하단의 ***Deploy (Manifest) Configuration*** 설정



- ***Mainfest Configuration – Manifest Source: Text*** 선택
- ***Manifest:***

```yaml
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  labels:
    app: recommendation-service
  name: recommendation-service-deployment-v2
  namespace: '${parameters.namespace}'
spec:
  replicas: 1
  selector:
    matchLabels:
      app: recommendation-service
  template:
    metadata:
      labels:
        app: recommendation-service
        version: v2
    spec:
      containers:
        - env:
            - name: SPRING_PROFILES_ACTIVE
              value: k8s
          image: dtlabs/recommendation-service:latest
          imagePullPolicy: Always
          name: recommendation-service
          ports:
            - containerPort: 8080
          resources:
            limits:
              cpu: '0.5'
              memory: 1Gi
            requests:
              cpu: '0.25'
              memory: 256Mi
```

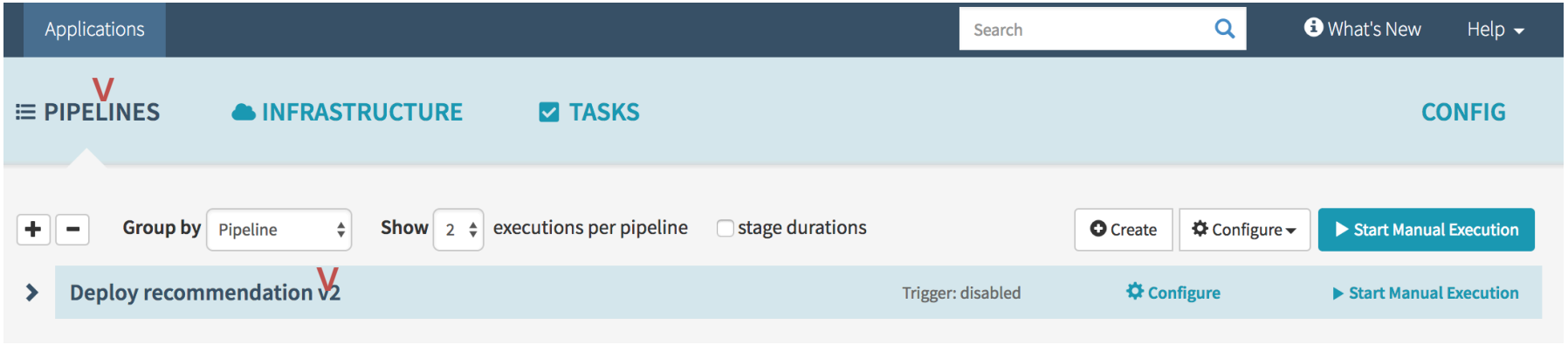
recommendation-service-deployment-v2.yaml

Copyright©2018 by DIGITAL LABS All rights reserved.

Docker hub triggering

5 Kubernetes에 Deployment 배포

✓ 생성된 파이프라인 확인



6 Docker Image 푸시

Docker Hub의 dtlabs/recommendation-service:2.0 Docker Image 푸시한다.

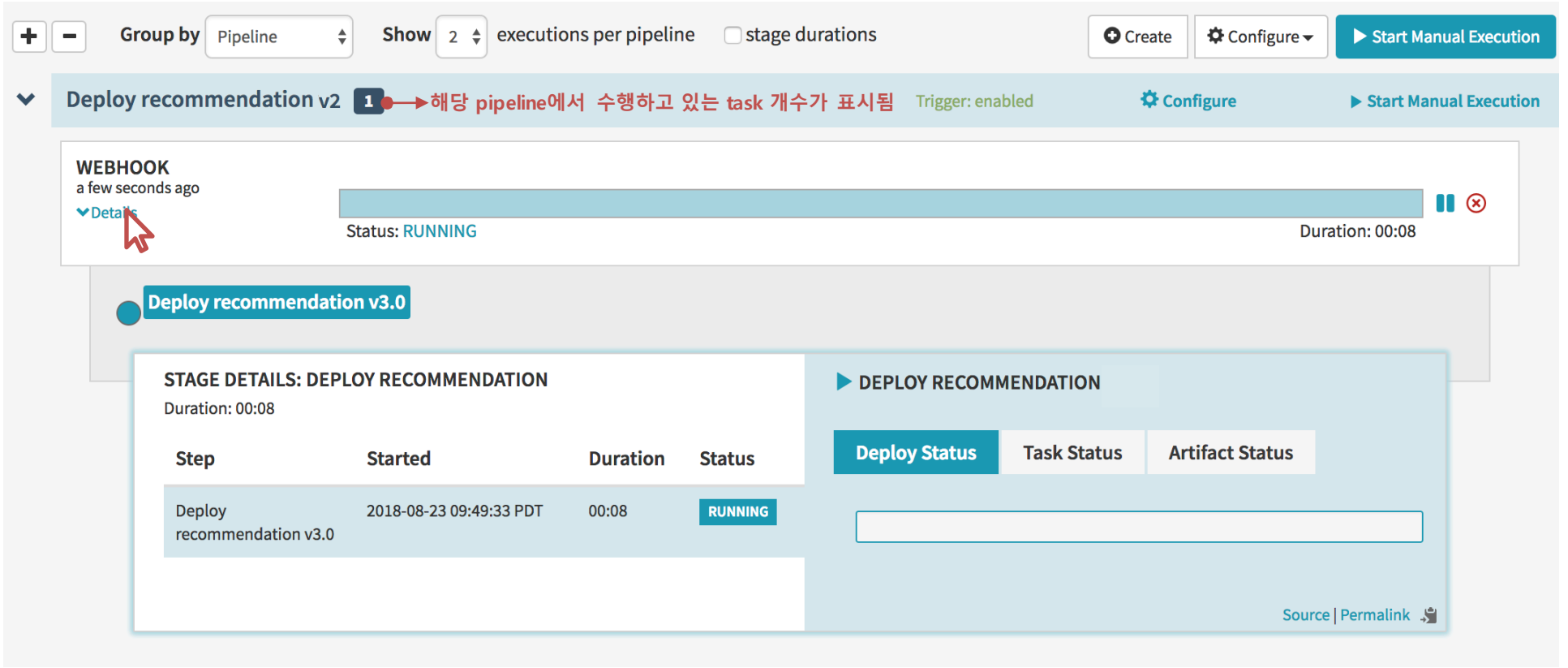
```
$ docker push dtlabs/recommendation-service:2.0
```

Copyright©2018 by DIGITAL LABS All rights reserved.

Docker hub triggering

Pipeline 실행 결과 확인

✓ Pipeline이 자동 시작하여 실행(RUNNING)

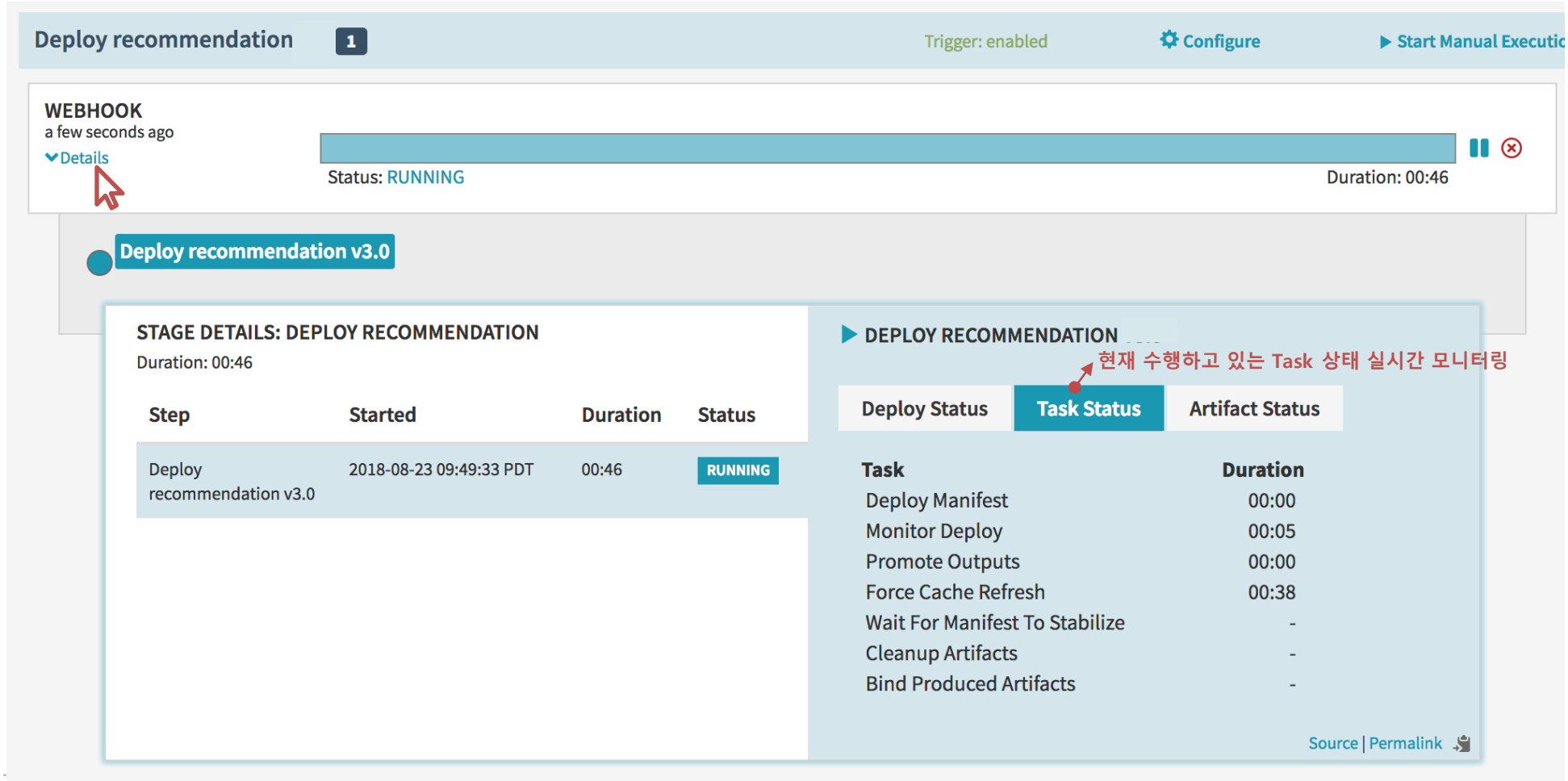


Copyright©2018 by DIGITAL LABS All rights reserved.

Docker hub triggering

Pipeline 실행 결과 확인

Details를 클릭하여 현재 실행 단계에 대한 상세 정보를 확인할 수 있다.

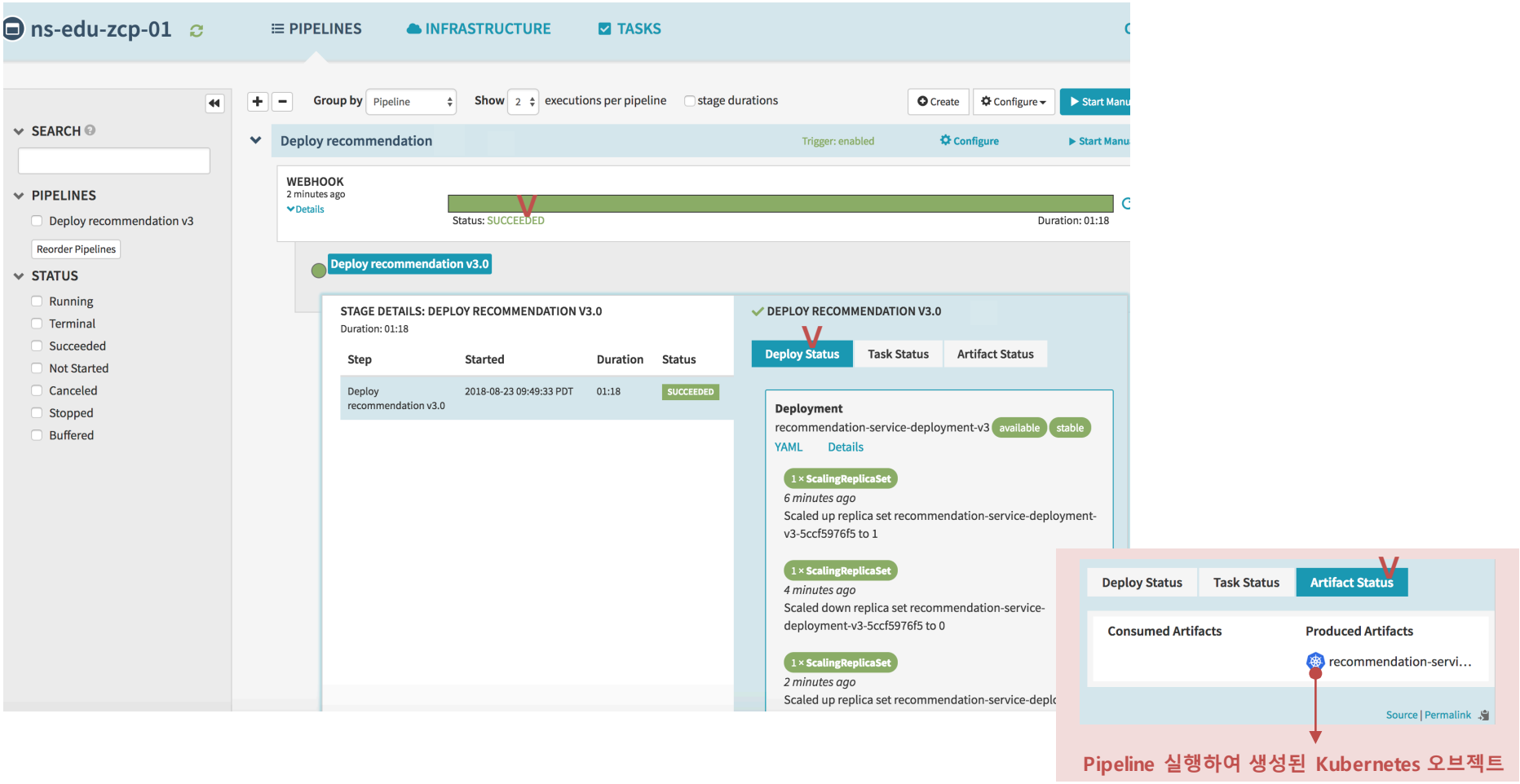


Copyright©2018 by DIGITAL LABS All rights reserved.

Docker hub triggering

Pipeline 실행 결과 확인

✓ Pipeline을 성공적으로 실행 완료되어 SUCCEEDED 표시된 결과 확인

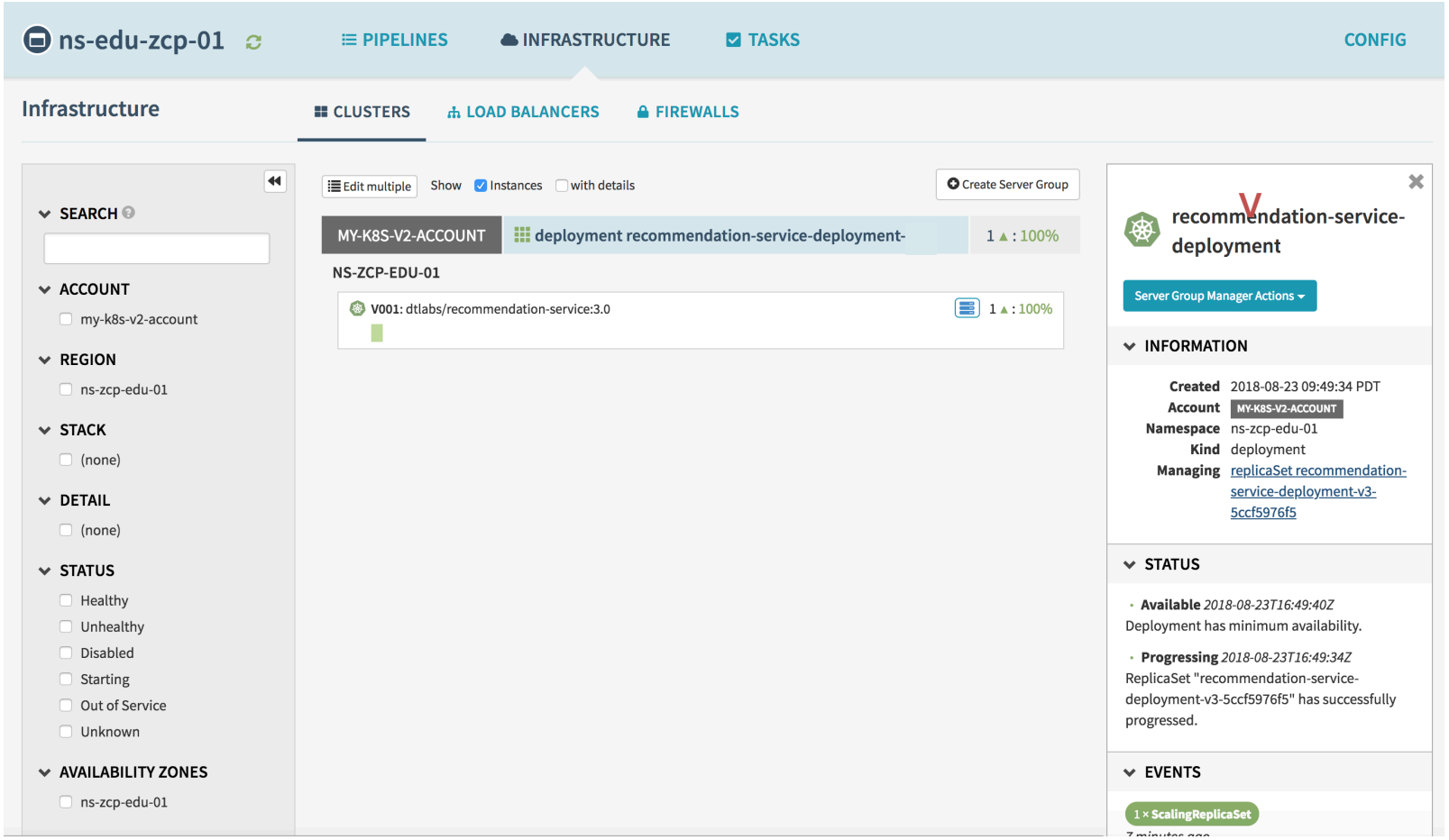


Copyright©2018 by DIGITAL LABS All rights reserved.

Docker hub triggering

Pipeline 실행 결과 확인

✓ Infrastructure에서 Workloads 확인



Copyright©2018 by DIGITAL LABS All rights reserved.

Docker hub triggering

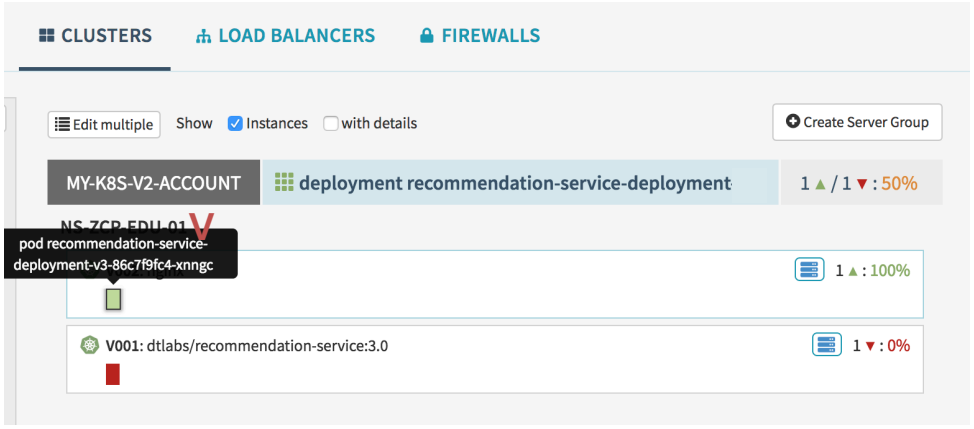
7 Docker Image Update

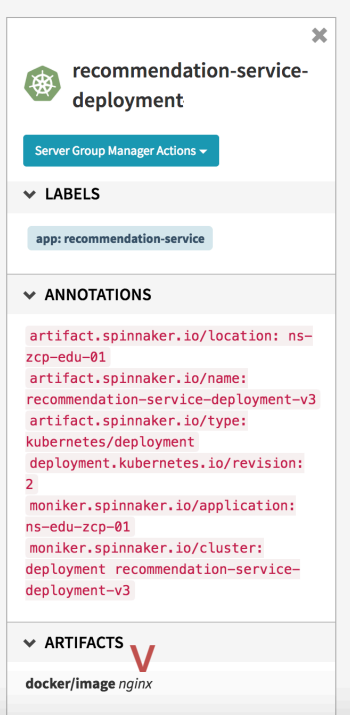
7 nginx Image로 deployment 업데이트

```
kubectl set image deployment/recommendation-service-deployment-v2 recommendation-service=nginx –n ${namespace name}
```

```
$ kubectl set image deployment/recommendation-service-deployment-v2 recommendation-service=nginx -n
${namespace name}
$ deployment.apps "recommendation-service-deployment-v2" image updated
```

✓ 업데이트 결과 확인





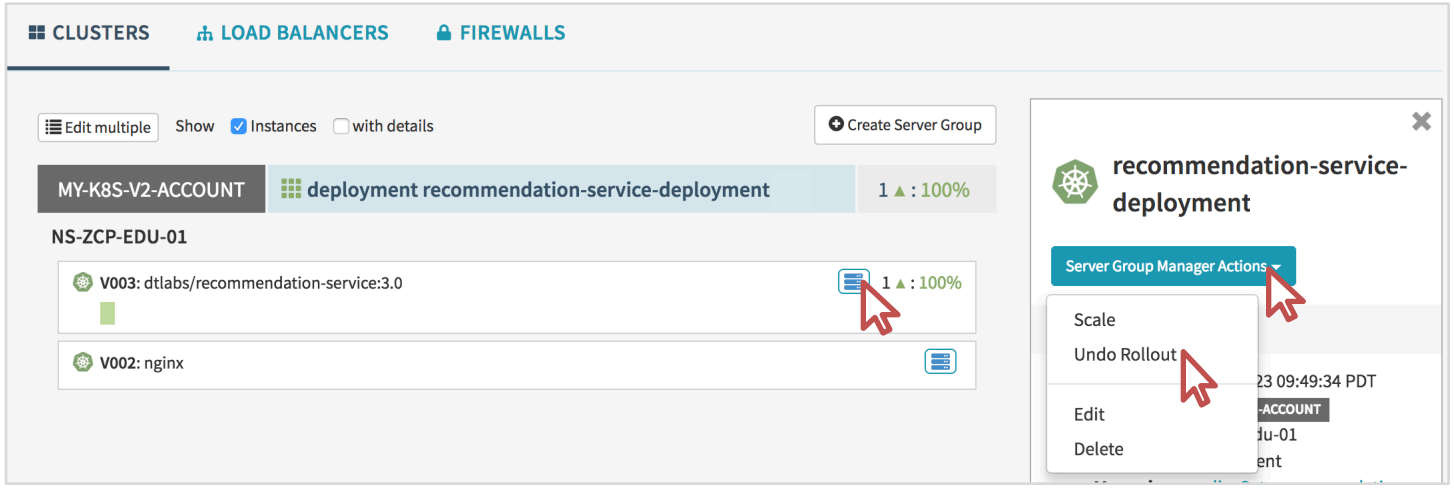
Copyright©2018 by DIGITAL LABS All rights reserved.

Docker hub triggering

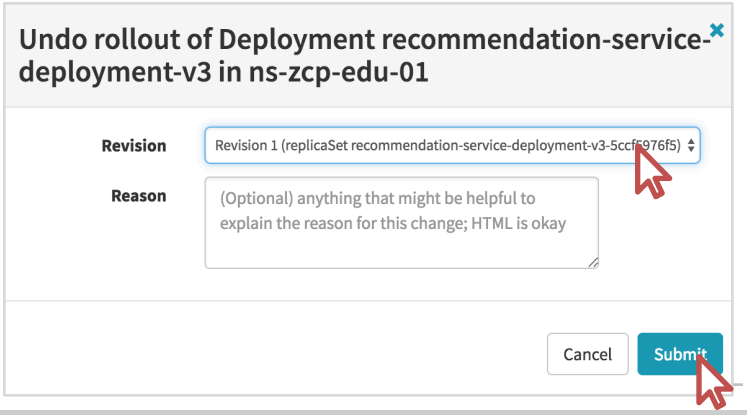
8 Rollout

최초 버전으로 rollout한다

8-1 Deployment 상세 뷰의 ***Server Group Manager Actions*** 클릭



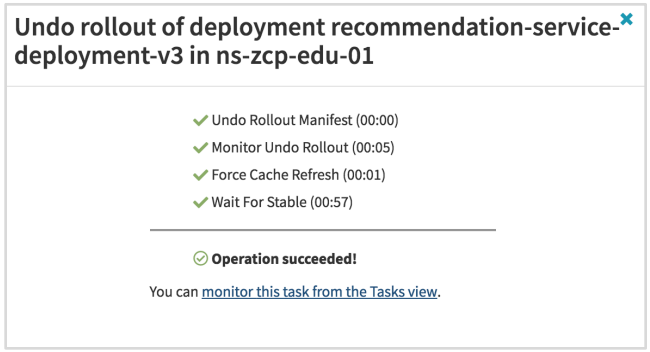
8-2 **Revision: Revision 1** 선택, ***Submit*** 버튼 클릭



Copyright©2018 by DIGITAL LABS All rights reserved.

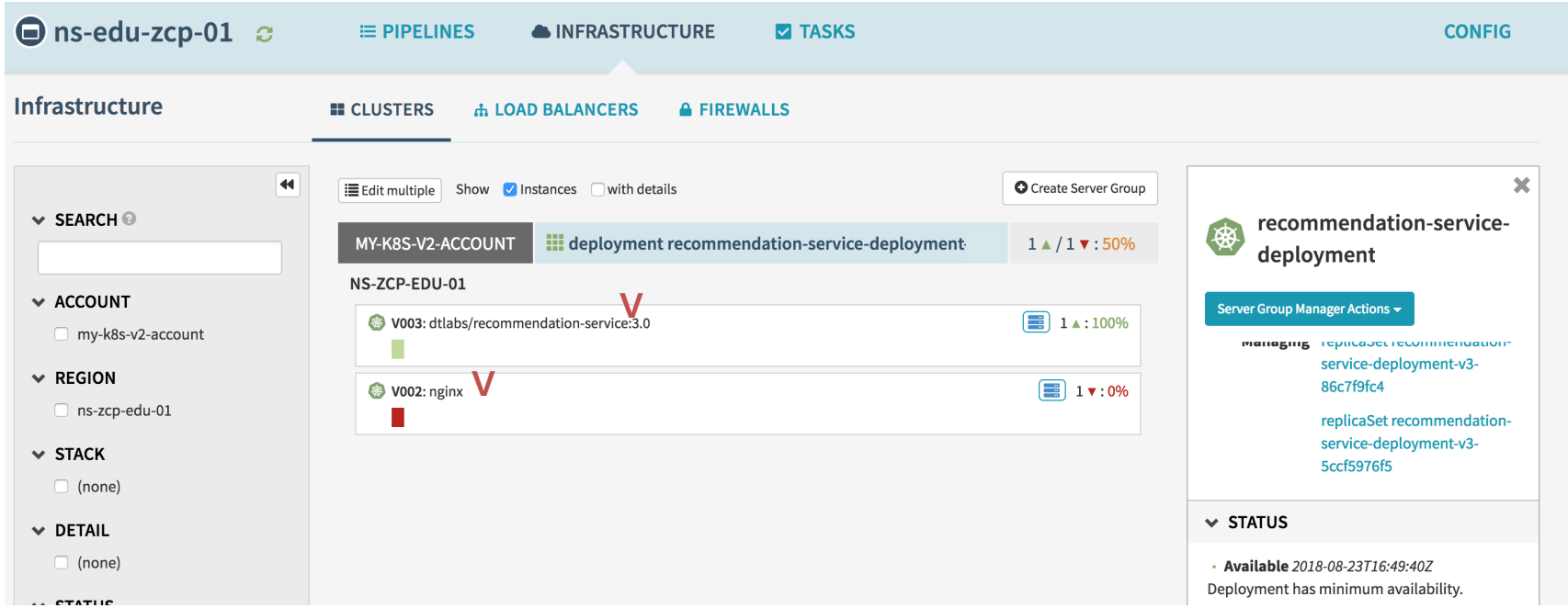
Docker hub triggering

8 Rollout



Rollout 실행 완료함

nginx가 적용된 pod가 없어지고 예전 Image가 적용된 pod가 생성됨을 확인



Copyright©2018 by DIGITAL LABS All rights reserved.

Hands-on: Deploy Pipeline 생성하기

01 Docker Hub Triggering를 통한 Deploy

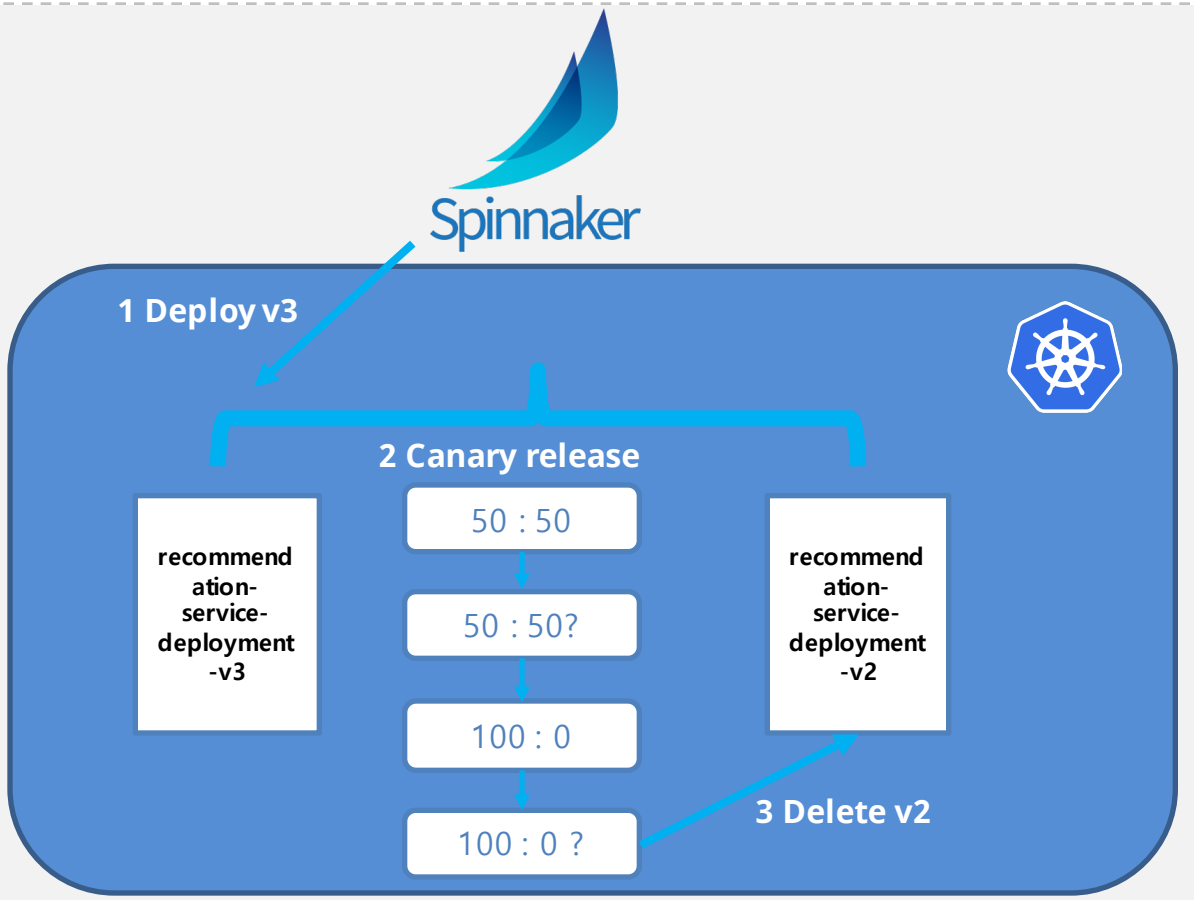
02 Canary release using istio

Copyrightⓒ2018 by DIGITAL LABS All rights reserved.

Canary release using istio

istio를 활용한 Canary release로 무중단 배포

시나리오



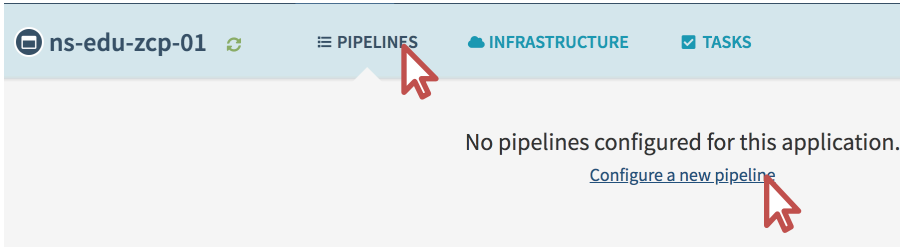
새 버전 애플리케이션을 canary release하고 기존 버전 삭제 (Parameter 사용)

Copyright©2018 by DIGITAL LABS All rights reserved.

Canary release using istio

1,2 Pipeline 생성, Parameters 적용

1-1 상단의 ***PIPELINES*** 클릭 – 하단의 **Configure a new pipeline** 클릭

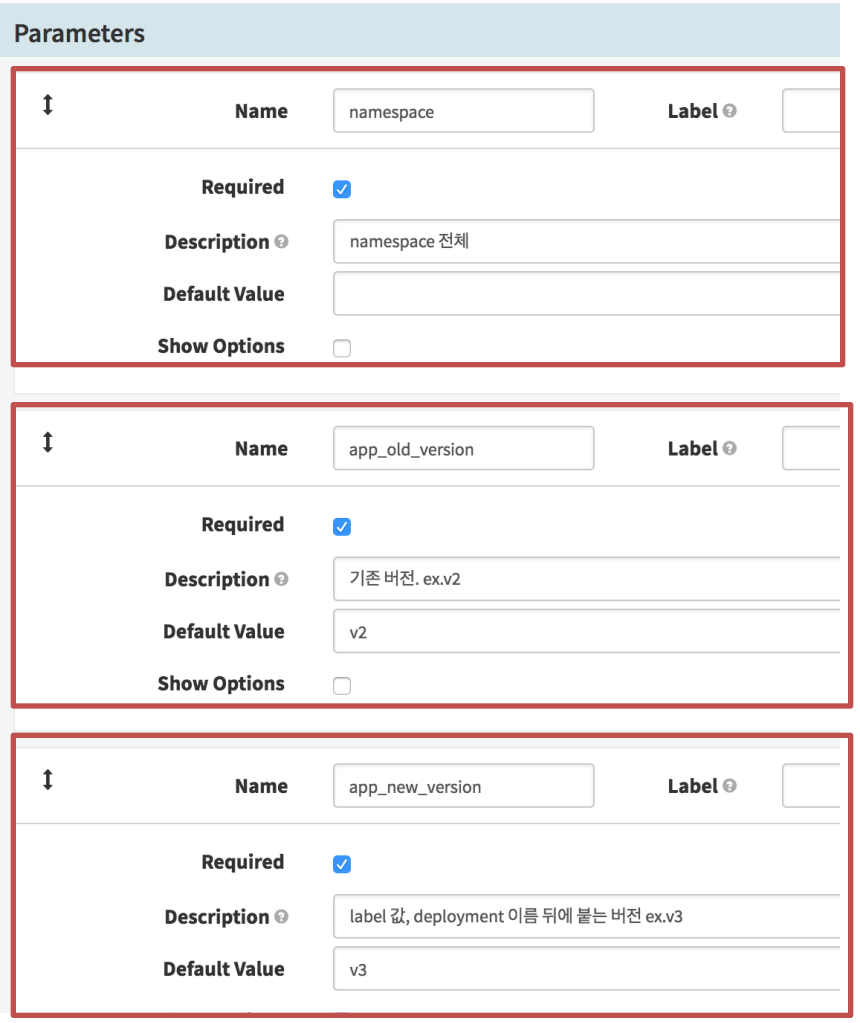


1-2 ***Type: Pipeline*** 선택
Pipeline Name: 'Canary release' 입력
Create 버튼 클릭

2 Parameters 적용
parameter를 설정한다.

Configuration stages의 **Parameters**에서 3개를 다음과 같이 설정

- ***Name:*** namespace
- ***Name***: app_old_versioin
- ***Name***: app_new_version



Copyright©2018 by DIGITAL LABS All rights reserved.

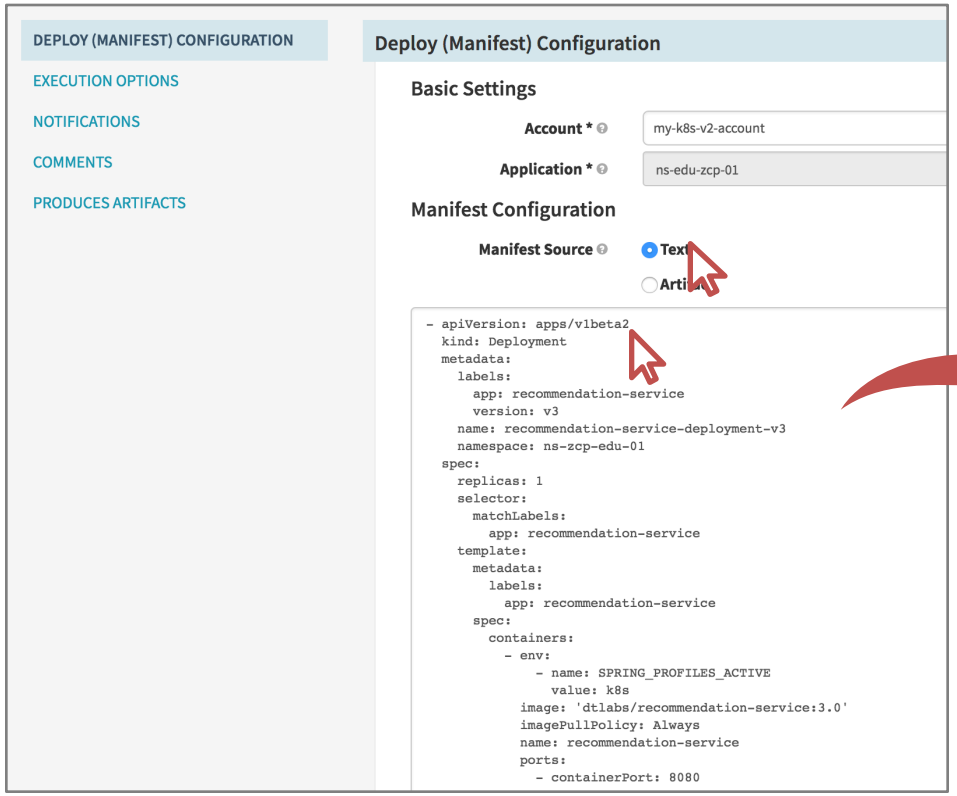
Canary release using istio

3 Kubernetes Deploy stage 생성하기

새로운 버전의 애플리케이션을 배포하는 stage를 생성한다.

3-1 상단의 ***PIPELINES*** 클릭 – 하단의 Configure a new pipeline
Stage Name: Deploy recommendation-new

3-2 하단의 ***Deploy (Manifest) Configuration*** 설정
recommendation-service-deployment.yaml 복사/붙여넣기




```yaml
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  labels:
    app: recommendation-service
  name: 'recommendation-service-deployment-${parameters.app_new_version}'
  namespace: '${parameters.namespace}'
spec:
  replicas: 1
  selector:
    matchLabels:
      app: recommendation-service
  template:
    metadata:
      labels:
        app: recommendation-service
        version: '${parameters.app_new_version}'
    spec:
      containers:
        - env:
            - name: SPRING_PROFILES_ACTIVE
              value: k8s
          image: dtlabs/recommendation-service:latest
          imagePullPolicy: Always
          name: recommendation-service
          ports:
            - containerPort: 8080
          resources:
            limits:
              cpu: '0.5'
              memory: 1Gi
            requests:
              cpu: '0.25'
              memory: 256Mi
```

Copyright©2018 by DIGITAL LABS All rights reserved.

Canary release using istio

4,5 istio rounting 5:5 적용

5:5 비율로 라우팅하기 위해 istio 오브젝트를 배포한다.

4-1 ***Add stage*** 클릭
- ***Stage Type***: ***Deploy (Manifest)*** 선택
  Depends On: Deploy recommendation - new 선택
  Stage Name: Deploy 5:5

4-2 배포할 Deployment의 매니페스트를 입력한다.
하단의 ***Deploy (Manifest) Configuration*** 설정

istio.yaml 복사/붙여넣기

5 Manual Judgment

사용자가 수동으로 5:5로 라우팅이 되는 지 확인하는 stage를 추가

Add stage 클릭 – ***Stage Type***: ***Manual Judgement***선택
Stage Name: 5:5?
Depends On: Deploy 5:5 선택

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: recommendation-service
  namespace: '${parameters.namespace}'
spec:
  host: recommendation-service
  subsets:
    - labels:
        version: '${parameters.app_old_version}'
      name: '${parameters.app_old_version}'
    - labels:
        version: '${parameters.app_new_version}'
      name: '${parameters.app_new_version}'
---
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: recommendation-service
  namespace: '${parameters.namespace}'
spec:
  hosts:
    - recommendation-service
  http:
    - route:
        - destination:
            host: recommendation-service
            subset: '${parameters.app_old_version}'
          weight: 50
        - destination:
            host: recommendation-service
            subset: '${parameters.app_new_version}'
          weight: 50
```

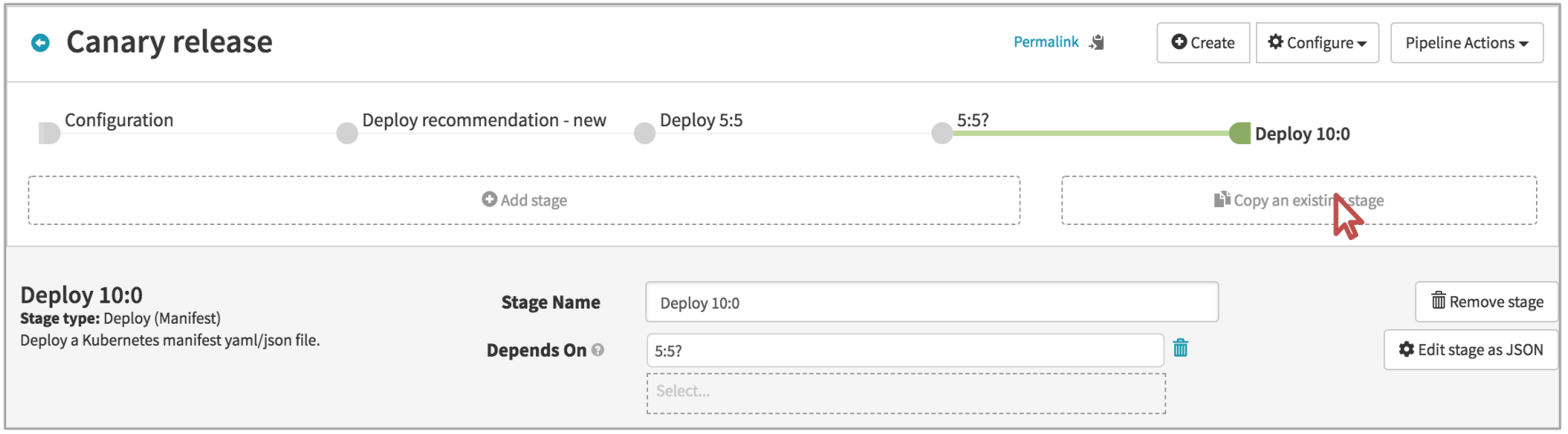
Copyright©2018 by DIGITAL LABS All rights reserved.

Canary release using istio

6 istio rounting 10:0 적용

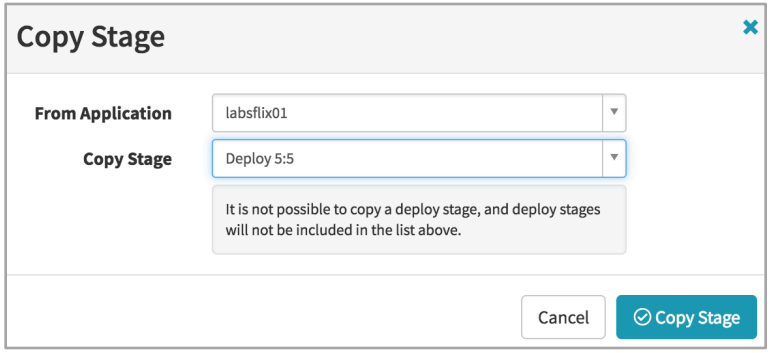
10:0 비율로 라우팅하기 위해 istio 오브젝트 배포한다.
이전에 만든 'Deploy 5:5' stages를 복사하여 사용한다.

6-1 **Copy an existing stage** 클릭



6-2 **Copy Stage** 팝업창에서 다음과 같이 선택

- **From Application:** 현재 application 선택
- **Copy Stage: Deploy 5:5**



Copyright©2018 by DIGITAL LABS All rights reserved.

Canary release using istio

6,7 istio rounting 10:0 적용

6-3 배포할 Deployment의 매니페스트를 수정한다.

하단의 ***Deploy (Manifest) Configuration*** 설정

subset: ${parameters.app_old_version}의 weight → 0 수정
subset: ${parameters.app_new_version} 의 weight → 100 수정

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: recommendation-service
  namespace: '${parameters.namespace}'
spec:
  hosts:
    - recommendation-service
  http:
    - route:
        - destination:
            host: recommendation-service
            subset: '${parameters.app_old_version}'
          weight: 0
        - destination:
            host: recommendation-service
            subset: '${parameters.app_new_version}'
          weight: 100
```

7 Manual Judgment

사용자가 수동으로 10:0로 라우팅이 되는 지 확인하는 stage를 추가

Add stage 클릭 – ***Stage Type***: ***Manual Judgement***선택
Stage Name: 10:0?
Depends On: **Deploy 10:0 선택**

Copyright©2018 by DIGITAL LABS All rights reserved.

Canary release using istio

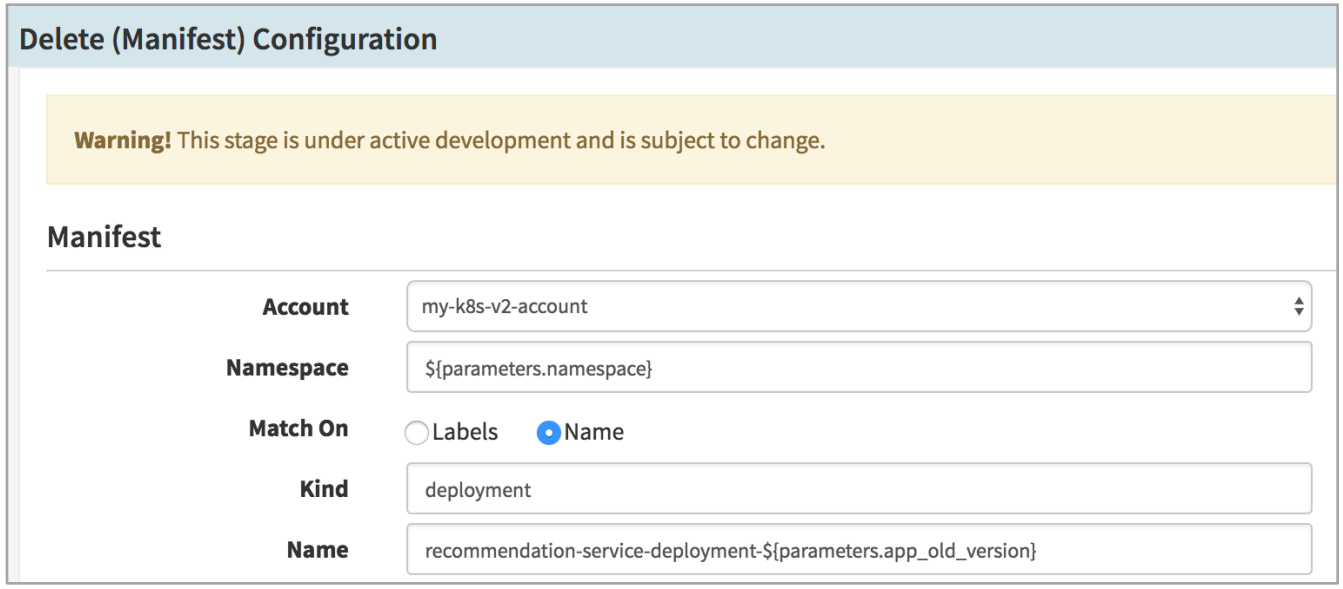
8 Deployment old version 삭제

예전 버전의 애플리케이션을 삭제한다.

8-1 ***Add stage*** 클릭
- ***Stage Type***: ***Deletey (Manifest)*** 선택
- ***Depends On***: 10:0?선택

8-2 하단의 Manifest를 다음과 같이 설정

- ***Namespace:*** ${parameters.namespace}
- ***Match On***: Name 선택
- ***Kind***: deployment
- ***Name***: recommendation-service-deployment-${parameters.app_old_version}

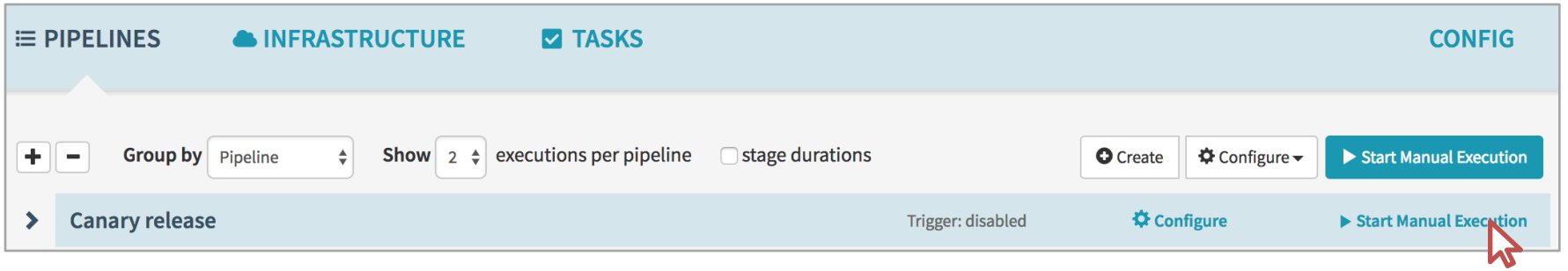


Copyright©2018 by DIGITAL LABS All rights reserved.

Canary release using istio

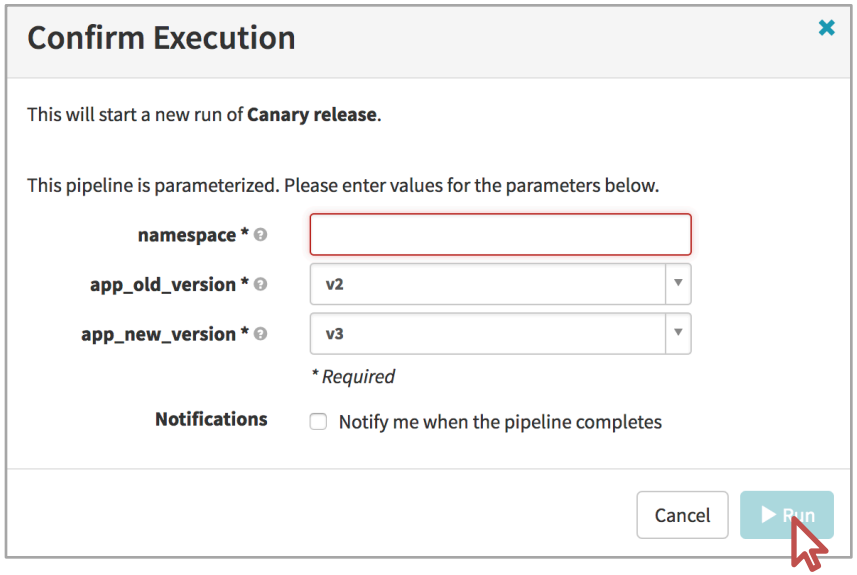
Pipeline 실행 결과 확인

1 Pipeline 화면에서 Canary release pipeline 수동 실행



2 Confirm Execution 설정
- **namespace:** 본인의 namespace명 전체 입력

3 **Run** 버튼 클릭



Copyright©2018 by DIGITAL LABS All rights reserved.

감사합니다