

Spinnaker를 활용한 배포 파이프라인 구성하기

2018.08.29



Table of Contents

01 | Spinnaker

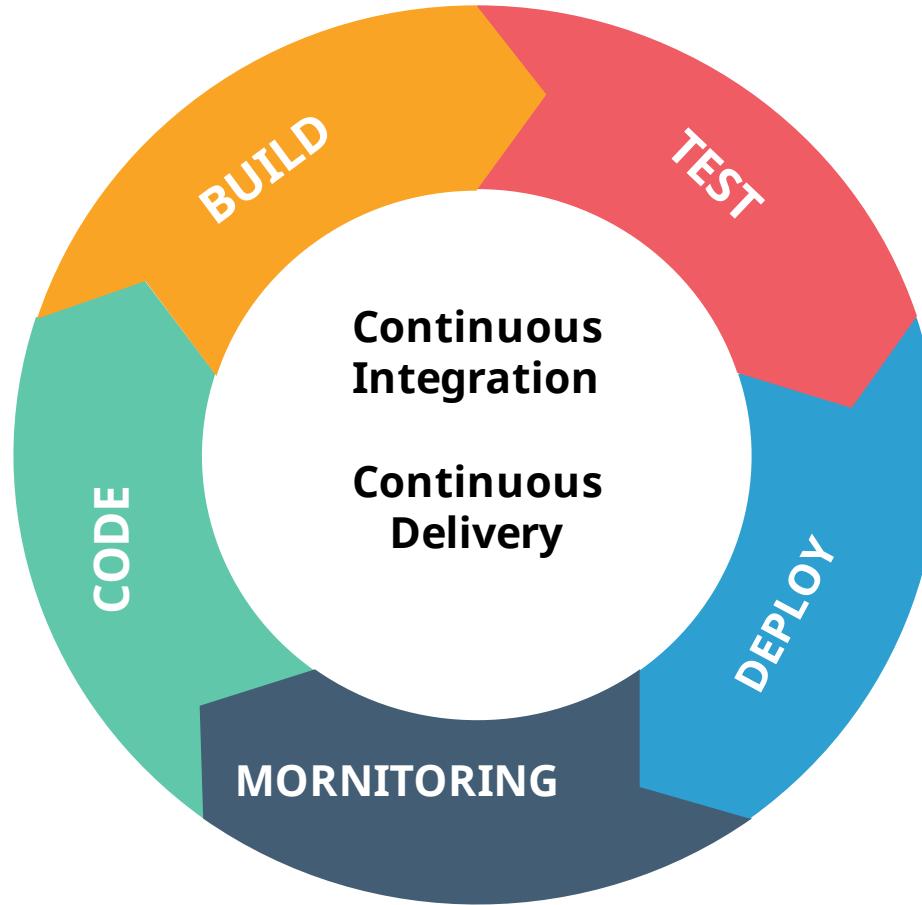
02 | Deploy Pipeline 구성하기

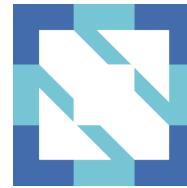
PART01. Spinnaker



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





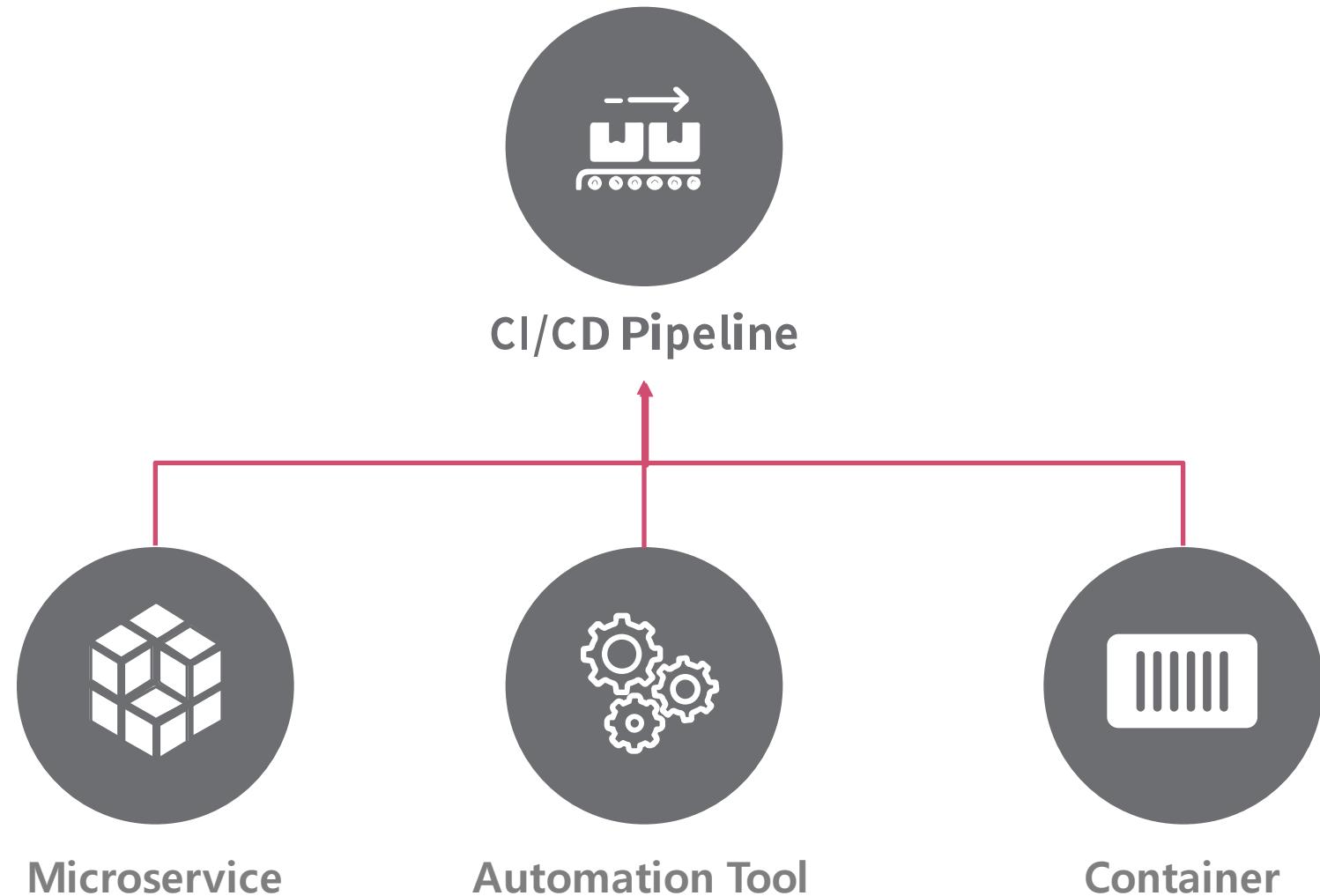
CLOUD NATIVE COMPUTING FOUNDATION

"

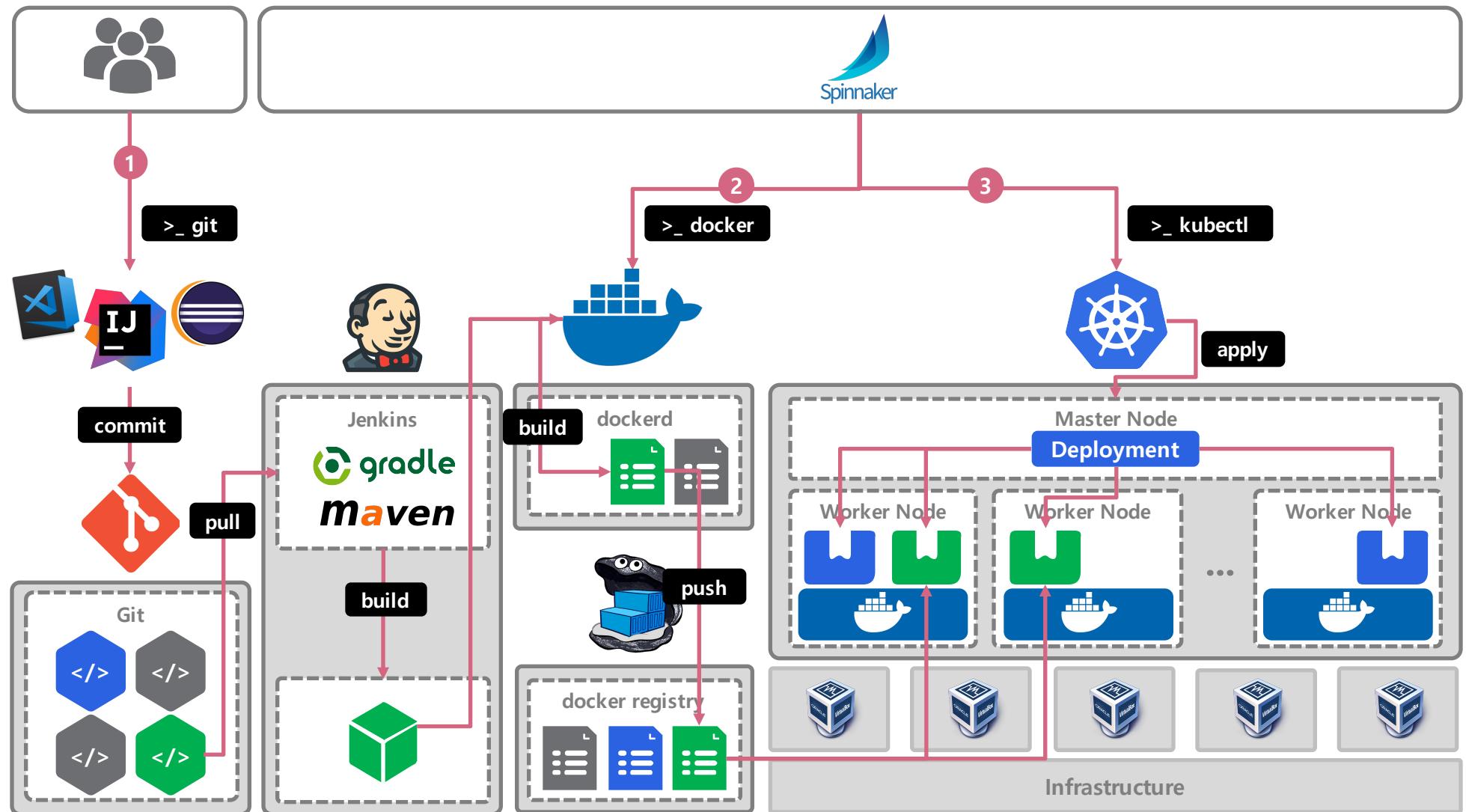
*Setup Continuous Integrating/Continuous Delivery
so that changes to your source code automatically result
in a new container being built, tested, and deployed
to staging and eventually, perhaps, to production*

Setup automated rollouts, roll backs and testing

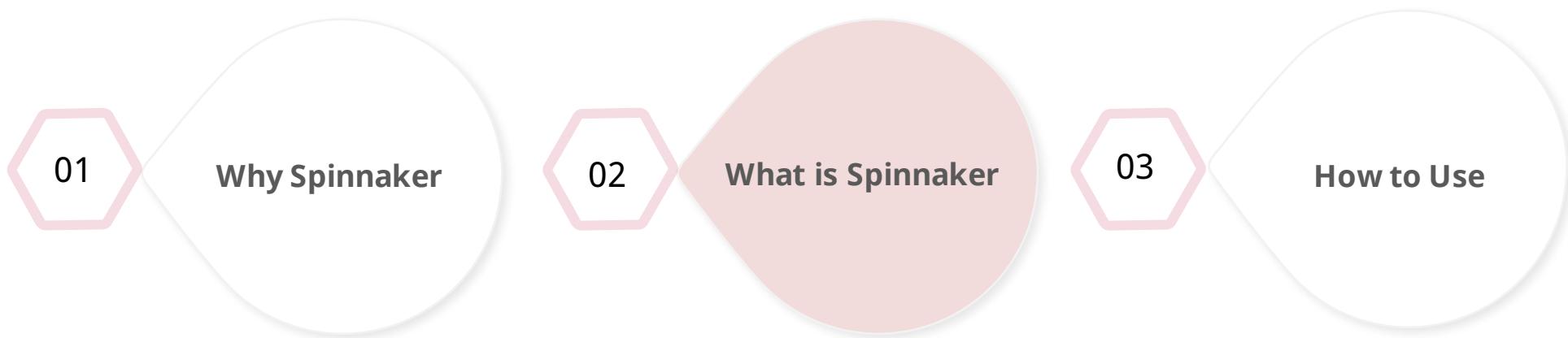
"



Continuous Integration, Continuous Delivery



PART01. Spinnaker



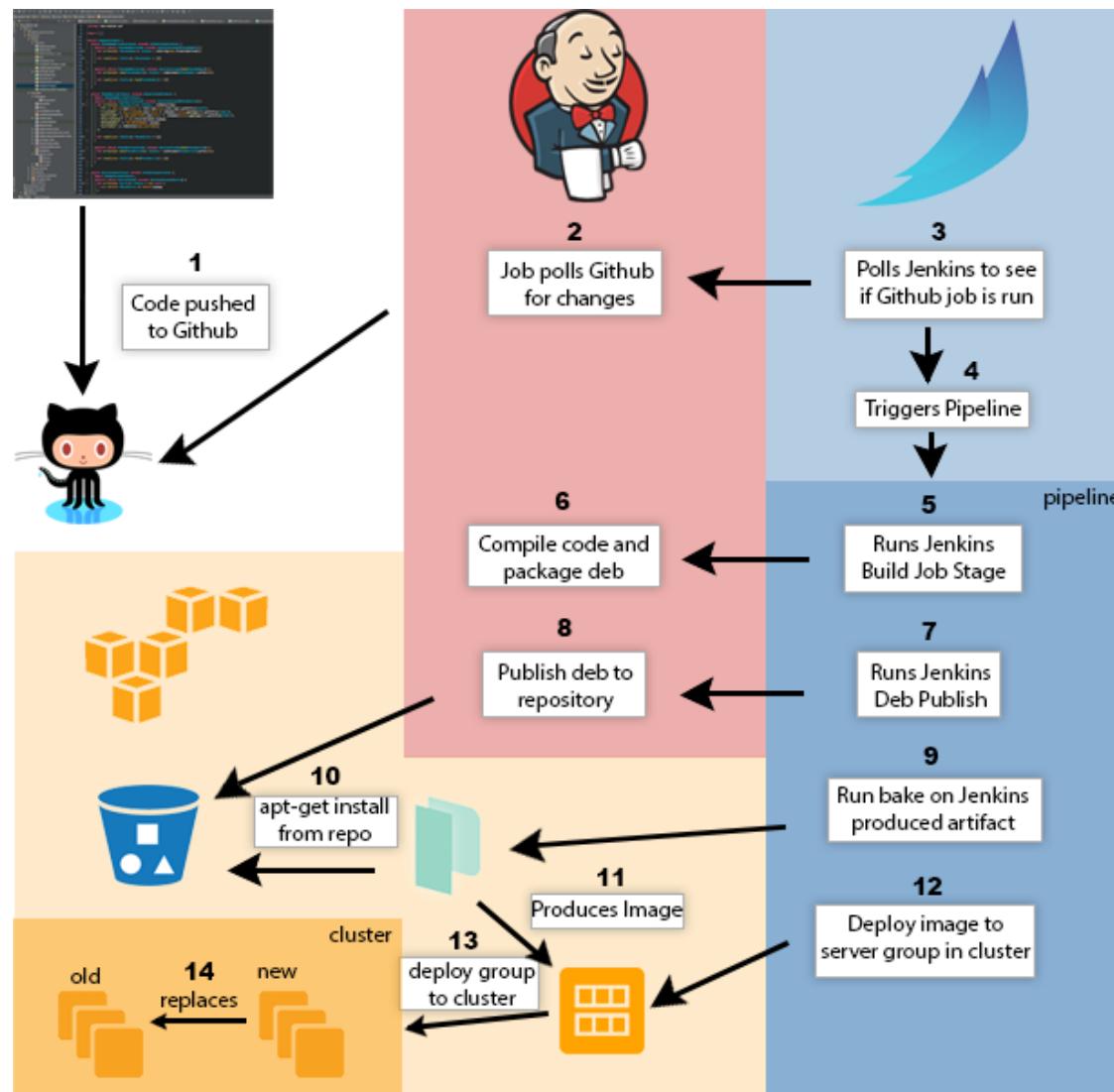
Spinnaker란

Continuous Delivery for Enterprise

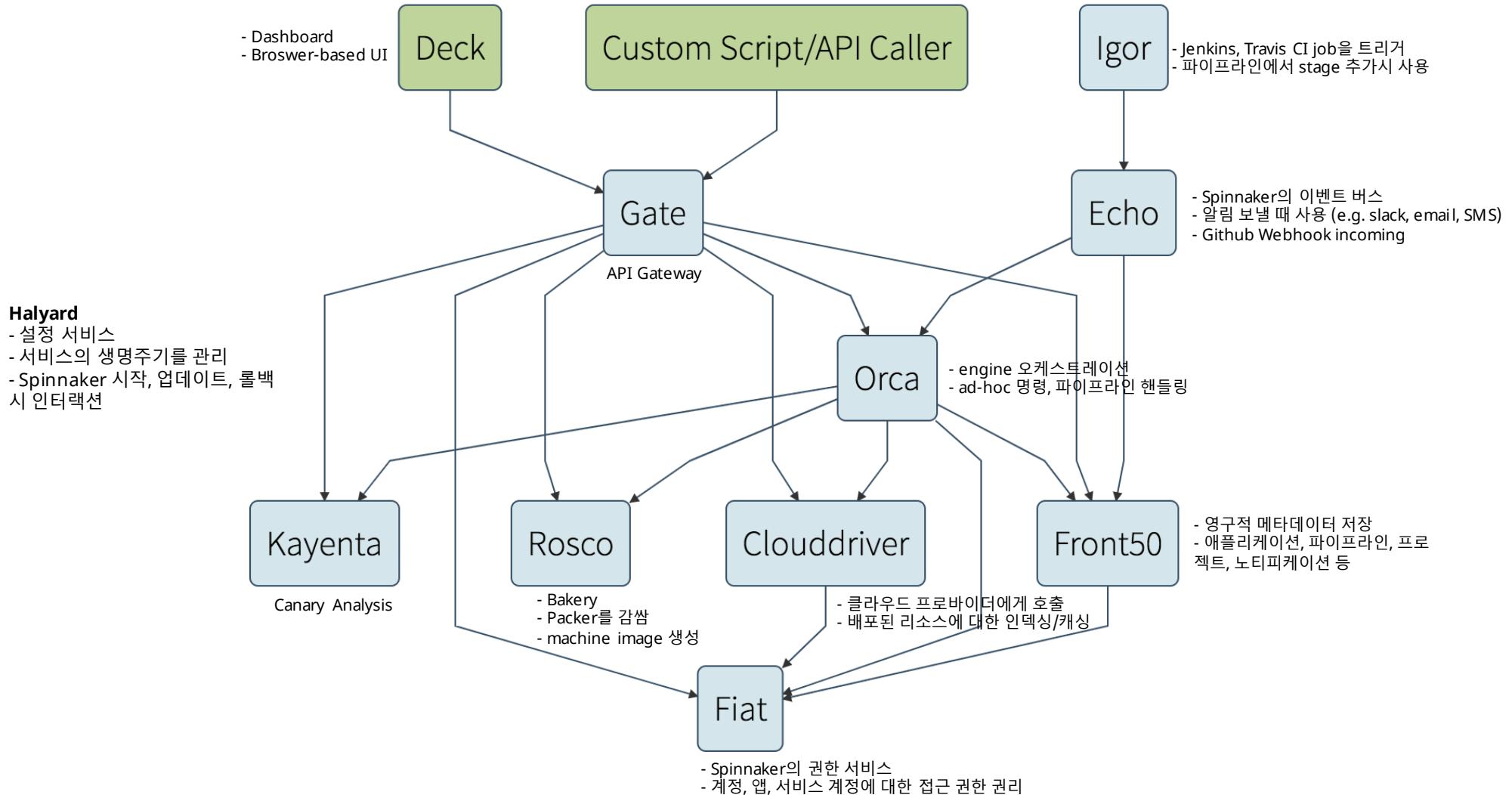
Fast, safe, repeatable deployments



Spinnaker란



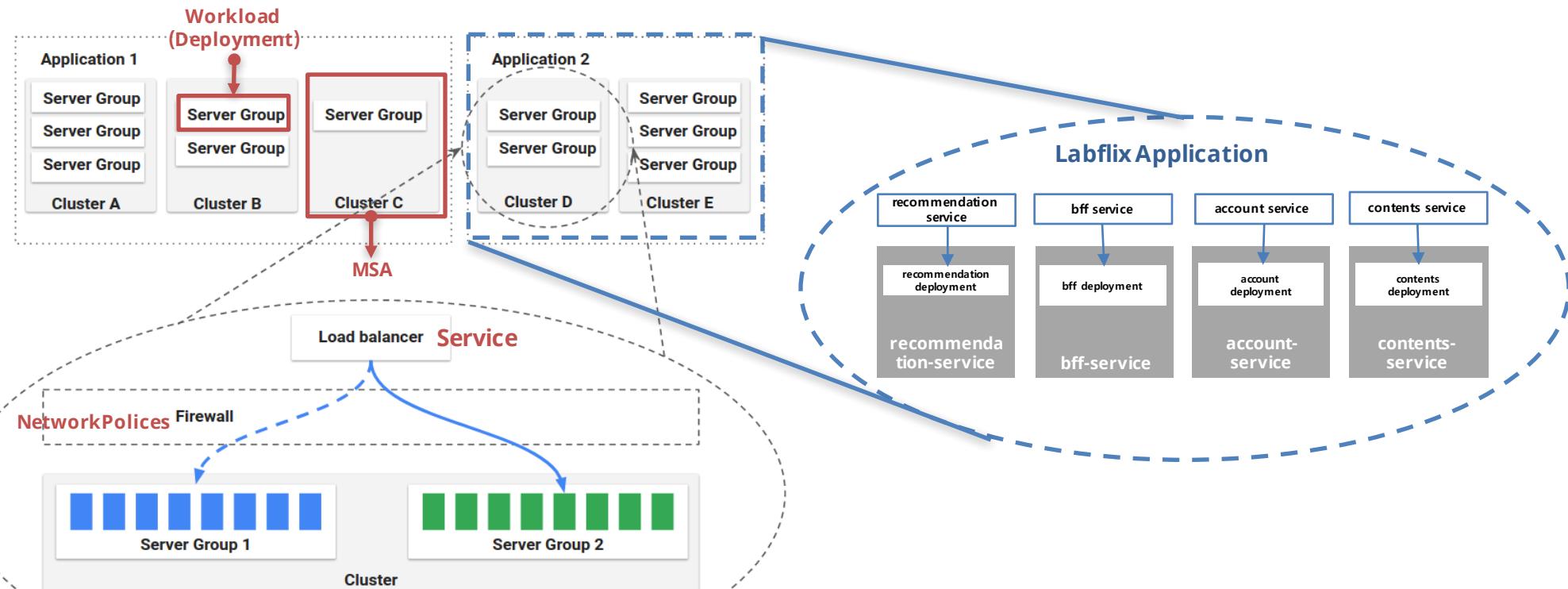
Spinnaker Architecture



Spinnaker Concept

1 Application Management: 클라우드에 있는 리소스를 조회, 관리

Application > Cluster > Server Group



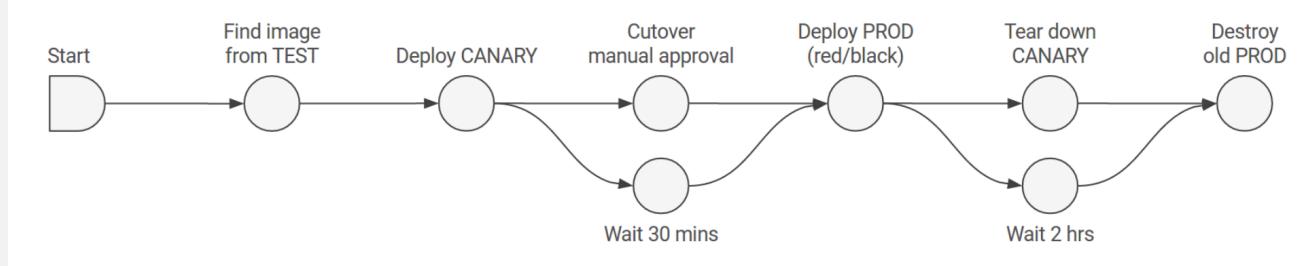
*Kubernetes 기준

Spinnaker Concept

2 Application Deployment: 지속적인 배포 워크 플로우를 구성하여 배포를 자동화

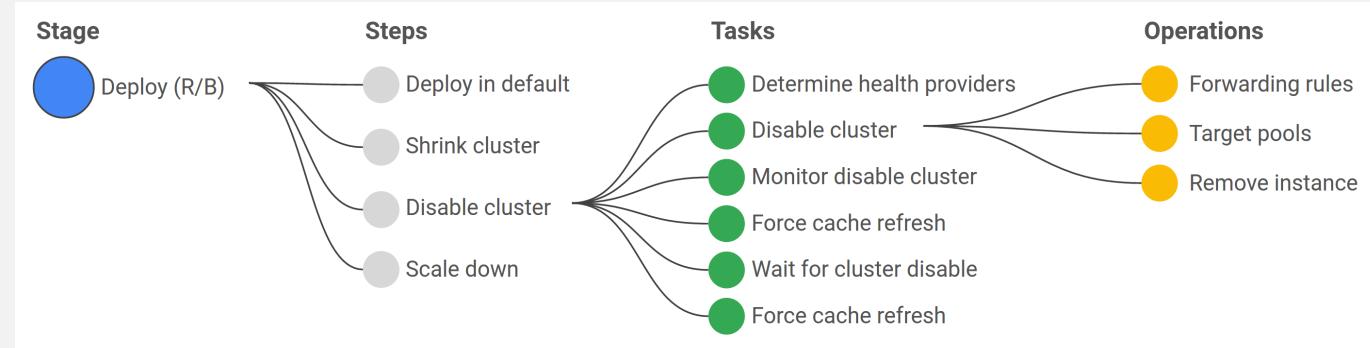
Pipeline

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



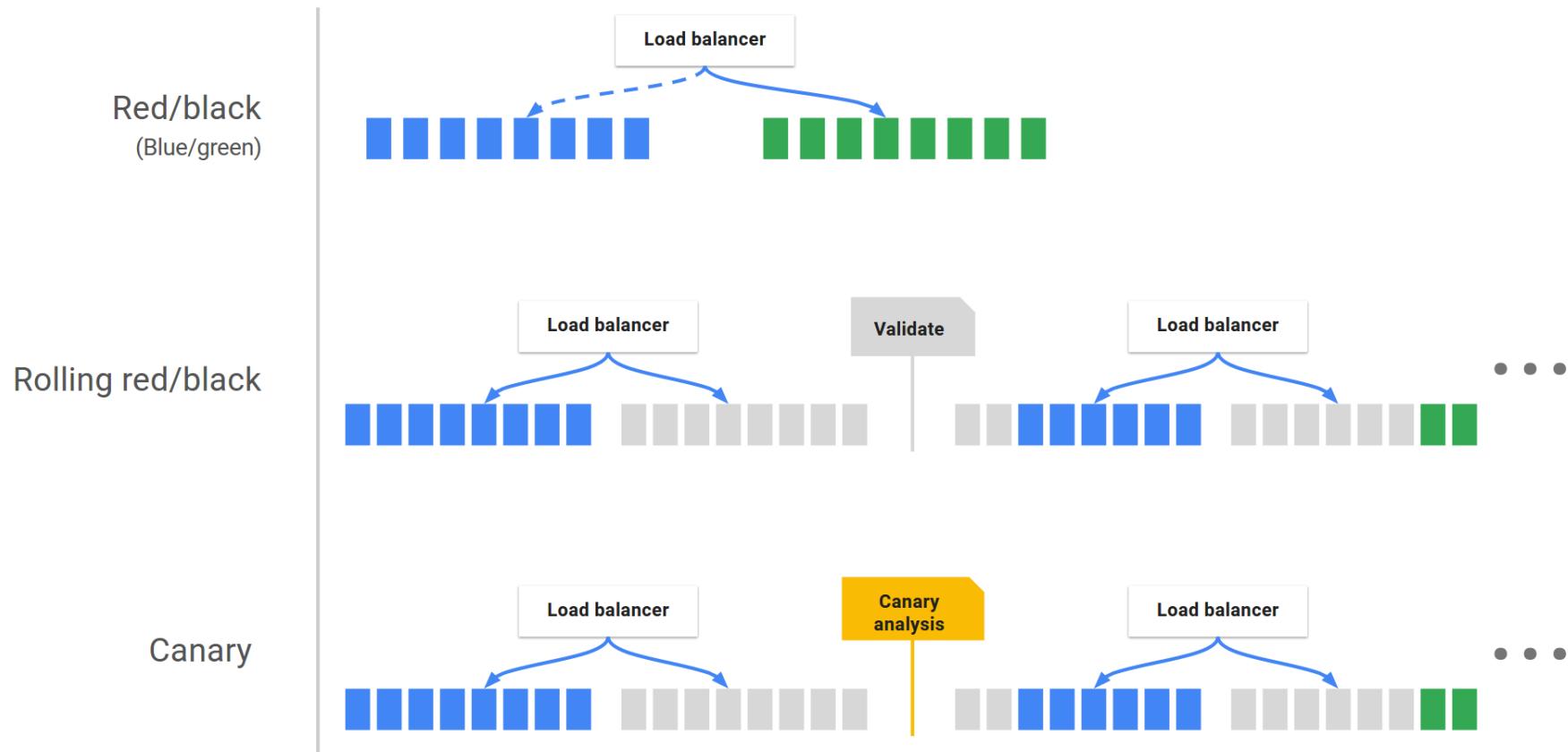
Stage

- Pipeline을 구성하는 가장 작은 단위
- 사용자가 수행하려는 동작을 설정



Spinnaker Concept

2 Application Deployment: 다양한 배포 전략을 기본적으로 제공



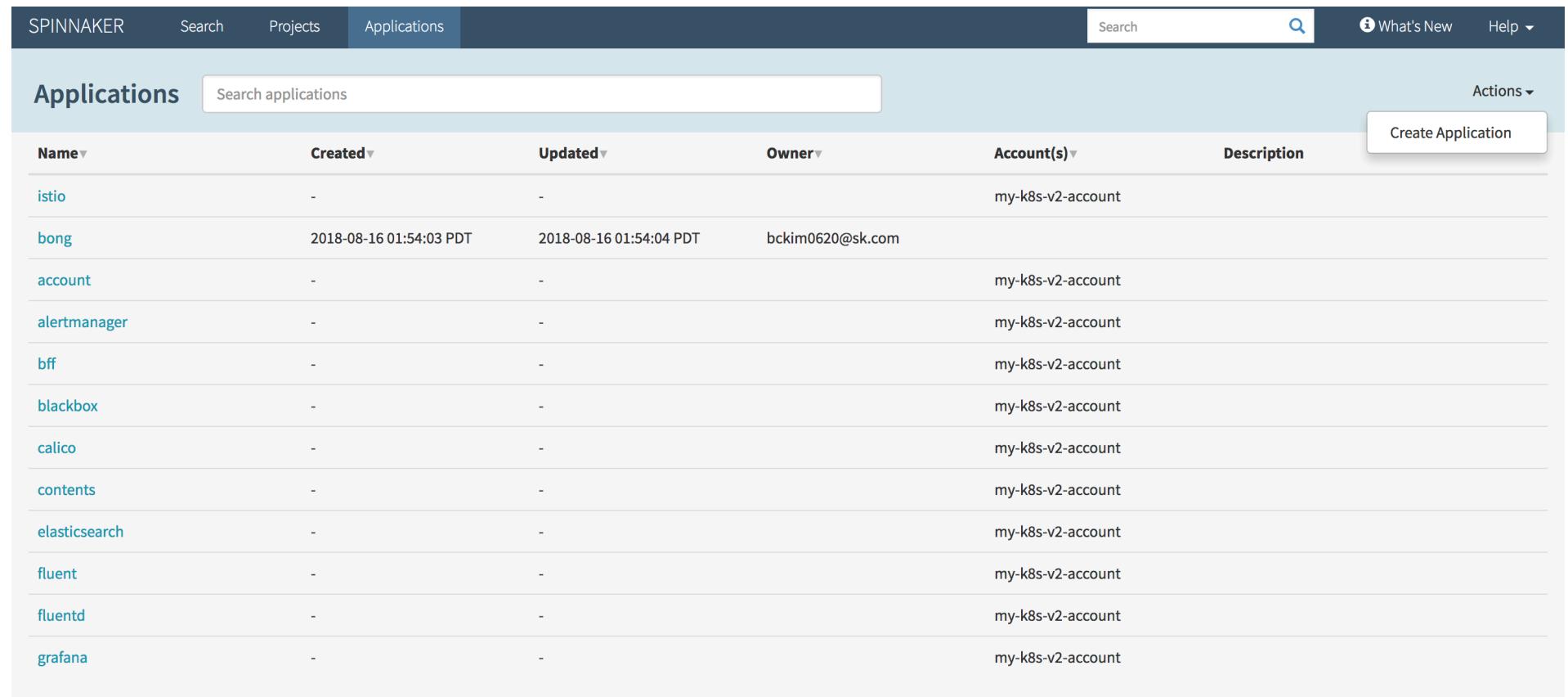
PART01. Spinnaker



Spinnaker 구성요소

Application

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



The screenshot shows the Spinnaker UI for managing applications. At the top, there's a navigation bar with links for SPINNAKER, Search, Projects, Applications (which is the active tab), and a search bar. To the right of the search bar are links for 'What's New' and 'Help'. Below the navigation is a header with 'Applications' and a search input field. On the far right of the header is a 'Create Application' button. The main area is a table listing various application names along with their creation and update times, owners, accounts, and descriptions.

Name	Created	Updated	Owner	Account(s)	Description	Action
istio	-	-		my-k8s-v2-account		
bong	2018-08-16 01:54:03 PDT	2018-08-16 01:54:04 PDT	bckim0620@sk.com			
account	-	-		my-k8s-v2-account		
alertmanager	-	-		my-k8s-v2-account		
bff	-	-		my-k8s-v2-account		
blackbox	-	-		my-k8s-v2-account		
calico	-	-		my-k8s-v2-account		
contents	-	-		my-k8s-v2-account		
elasticsearch	-	-		my-k8s-v2-account		
fluent	-	-		my-k8s-v2-account		
fluentd	-	-		my-k8s-v2-account		
grafana	-	-		my-k8s-v2-account		

Spinnaker 구성요소

Pipeline

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

PIPELINES

- bff-with-trigger
- canary-with-istio
- deploy all backend v1
- delete all backend
- canary-maual-execution

STATUS

- Running
- Terminal
- Succeeded
- Not Started
- Canceled
- Stopped
- Buffered

bff-with-trigger

Trigger: enabled

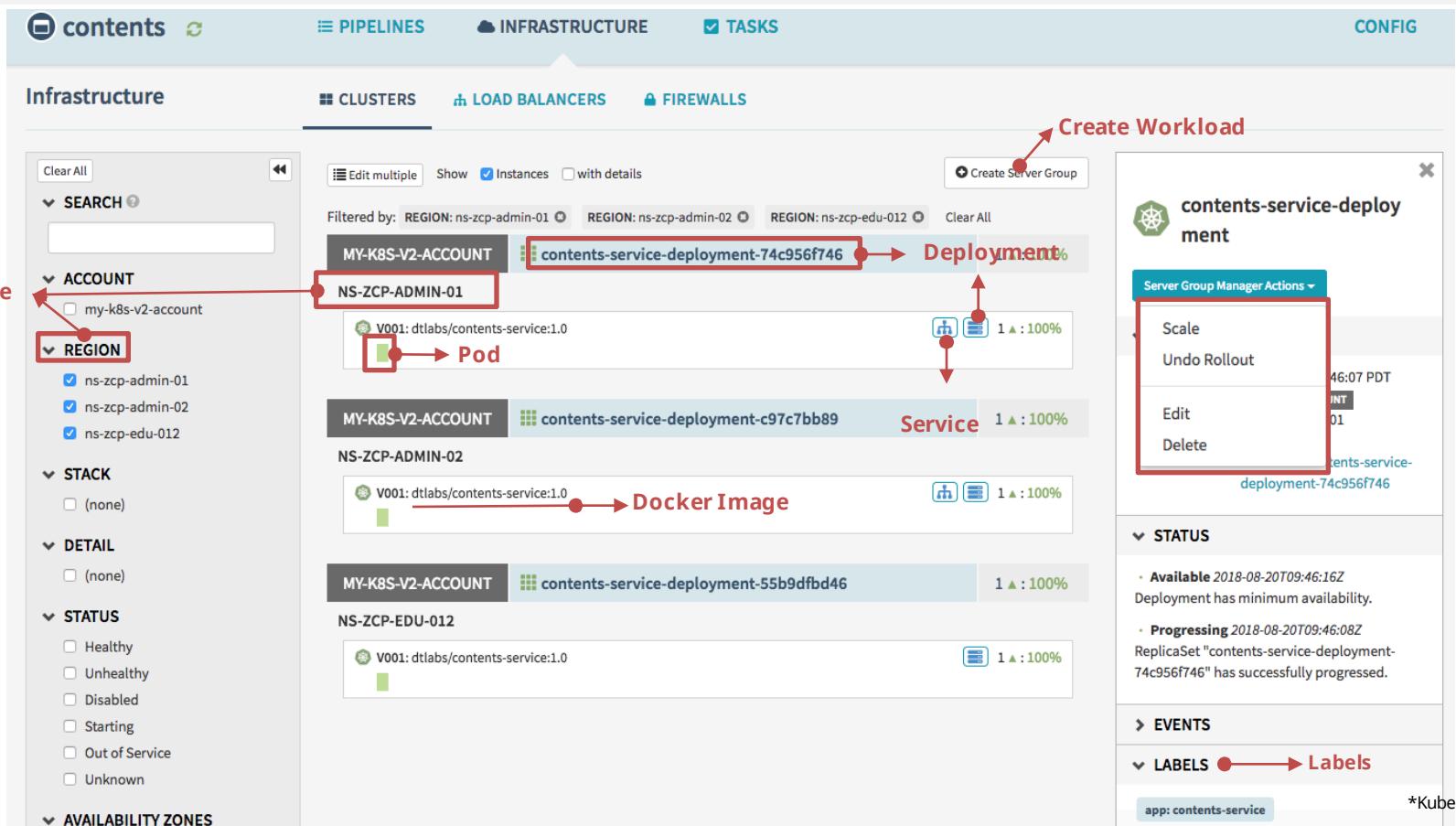
Execution	Version	Status	Duration
index.docker.io/dtlabs/bff-service	latest	SUCCEEDED	01:15
index.docker.io/dtlabs/bff-service	2.0	SUCCEEDED	02:12

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



Spinnaker 구성요소

Infrastructure

Load Balancers Kubernetes의 Service Controller

Firewall Kubernetes의 NetworkPolicy

The screenshot shows the Spinnaker Infrastructure page with the following details:

- Header:** bff, PIPELINES, INFRASTRUCTURE (selected), TASKS, CONFIG.
- Sub-Header:** Infrastructure, CLUSTERS, LOAD BALANCERS (selected), FIREWALLS.
- Search:** SEARCH, ACCOUNT (my-k8s-v2-account), REGION (ns-zcp-admin-02, ns-zcp-edu-01, ns-zcp-admin-01, ns-zcp-edu-02), STACK ((none), service), DETAIL ((none)), INSTANCE STATUS (Healthy).
- Load Balancers:** Show Server Groups (checked) and Instances (unchecked). A list of server groups and their instances:
 - MY-K8S-V2-ACCOUNT
 - service bff-service
 - NS-ZCP-ADMIN-02: 1 ▲ : 100%
 - replicaSet bff-service-deployment-d8777b46c
 - replicaSet bff-service-deployment-85858cf986
 - NS-ZCP-EDU-01: 2 ▲ : 100%
 - replicaSet bff-service-deployment-v2-794d8d8c4f
 - replicaSet bff-service-deployment-v1-59c969d6fb
 - NS-ZCP-EDU-02: 2 ▲ : 100%
 - replicaSet bff-service-deployment-v2-6b99474956
 - replicaSet bff-service-deployment-v1-8589f58cdf
 - NS-ZCP-ADMIN-01: 2 ▲ : 100%
 - replicaSet bff-service-deployment-v1-7f68d5c6bf
- Modal (bff-service-deployment-v1-59c969d6fb):**
 - Server Group Actions:**
 - INFORMATION:** Created: 2018-08-21 21:31:59 PDT, Account: MY-K8S-V2-ACCOUNT, Namespace: ns-zcp-edu-01, Kind: replicaSet, Controller: Deployment bff-service-deployment-v1.
 - EVENTS:**
 - LABELS:** app: bff-service, pod-template-hash: 1575258296.

Spinnaker 구성요소

Task

해당 Application에서 동작하고 있는 task를 한눈에 관리할 수 있는 뷰

contents **PIPLINES** **INFRASTRUCTURE** **TASKS** **CONFIG**

Task History

Task	Account	Region	Progress	Started	Ended	Running Time	User	Actions
> Save pipeline 'bff-with-trigger'			<div style="width: 100%; background-color: #6aa84f;"></div>	2018-08-23 00:23:37 PDT	2018-08-23 00:23:40 PDT	00:03	anonymous	
▼ Save pipeline 'bff-with-trigger'			<div style="width: 100%; background-color: #6aa84f;"></div>	2018-08-21 22:57:39 PDT	2018-08-21 22:57:42 PDT	00:02	anonymous	
✓ Save Pipeline				2018-08-21 22:57:39 PDT	2018-08-21 22:57:42 PDT	00:02		
✓ Wait For Pipeline Save				2018-08-21 22:57:42 PDT	2018-08-21 22:57:42 PDT	00:00		
Source Permalink								
> Save pipeline 'bff-with-trigger'			<div style="width: 100%; background-color: #6aa84f;"></div>	2018-08-21 22:57:37 PDT	2018-08-21 22:57:40 PDT	00:02	anonymous	
> Save pipeline 'canary-maul-excution'			<div style="width: 100%; background-color: #6aa84f;"></div>	2018-08-20 23:31:20 PDT	2018-08-20 23:31:23 PDT	00:03	anonymous	
> Save pipeline 'Lㅁ'			<div style="width: 100%; background-color: #6aa84f;"></div>	2018-08-20 22:59:40 PDT	2018-08-20 22:59:43 PDT	00:03	anonymous	
> Save pipeline 'delete all backend'			<div style="width: 100%; background-color: #6aa84f;"></div>	2018-08-20 01:26:46 PDT	2018-08-20 01:26:49 PDT	00:02	anonymous	
> Save pipeline 'delete all backend'			<div style="width: 100%; background-color: #6aa84f;"></div>	2018-08-20 01:23:37 PDT	2018-08-20 01:23:40 PDT	00:02	anonymous	
> Save pipeline 'delete all backend'			<div style="width: 100%; background-color: #6aa84f;"></div>	2018-08-20 01:22:08 PDT	2018-08-20 01:22:11 PDT	00:02	anonymous	
> Save pipeline 'delete all backend'			<div style="width: 100%; background-color: #6aa84f;"></div>	2018-08-20 01:20:52 PDT	2018-08-20 01:20:54 PDT	00:02	anonymous	
> Save pipeline 'delete all backend'			<div style="width: 100%; background-color: #6aa84f;"></div>	2018-08-20 01:18:58 PDT	2018-08-20 01:19:00 PDT	00:02	anonymous	
> Save pipeline 'delete all backend'			<div style="width: 100%; background-color: #6aa84f;"></div>	2018-08-20 01:18:23 PDT	2018-08-20 01:18:26 PDT	00:03	anonymous	
> Save pipeline 'delete all backend'			<div style="width: 100%; background-color: #6aa84f;"></div>	2018-08-20 01:14:19 PDT	2018-08-20 01:14:22 PDT	00:02	anonymous	
> Save pipeline 'delete all backend'			<div style="width: 100%; background-color: #6aa84f;"></div>	2018-08-20 01:13:31 PDT	2018-08-20 01:13:34 PDT	00:02	anonymous	
> Save pipeline 'delete all backend'			<div style="width: 100%; background-color: #6aa84f;"></div>	2018-08-20 01:12:59 PDT	2018-08-20 01:13:02 PDT	00:02	anonymous	

Spinnaker 구성요소

Project

Project

사용자가 관리하려는 Cluster, Application, Pipeline의 집합

Configure Project

PROJECT ATTRIBUTES

- APPLICATIONS
- CLUSTERS
- PIPELINES

Project Attributes

Project Name	<input type="text"/>
Owner Email	<input type="text"/> Enter an email address

Applications

Select...

Clusters

Applications	Account	Stack	Detail
-	-	-	Add Cluster

Pipelines

(Add an application)

SPINNAKER
Search
Projects
Applications

dna / Project Dashboard ▾

Application Status

Filter by region / namespace ▾

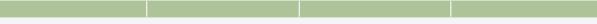
MY-K8S-V2-ACCOUNT		3 Applications 23 Instances 13 ▲ / 10 ▼ : 56%						
		Last Push	ns-zcp-admin-01	ns-zcp-admin-02	ns-zcp-edu-01	ns-zcp-edu-012	ns-zcp-edu-02	spinnaker
BFF	#0	20 hours ago	1 ▲ : 100%	1 ▲ : 100%	2 ▲ : 100%	1 ▲ : 100%	2 ▲ : 100%	-
CONTENTS	#0	14 hours ago	1 ▲ : 100%	1 ▲ : 100%	-	1 ▲ : 100%	-	10 ▼ : 0%
RECOMMENDATION	#0	19 hours ago	1 ▲ : 100%	1 ▲ : 100%	-	1 ▲ : 100%	-	-

Pipeline Status

CONTENTS: bff-with-trigger (started 2018-08-22 03:52:55 PDT)



CONTENTS: canary-with-istio (started 2018-08-17 04:15:18 PDT)



Pipeline 생성

1. Configuration 설정

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

The screenshot shows the Spinnaker interface for configuring a pipeline named 'bff-with-trigger'. The 'Configuration' tab is active, showing a flow diagram with stages: 'Deploy bff Deployment' followed by 'deploy bff'. There are buttons for 'Add stage' and 'Permalink'. Below the diagram, under 'Concurrent Executions', there are two checkboxes: 'Disable concurrent pipeline executions (only run one at a time)' (checked) and 'Do not automatically cancel pipelines waiting in queue'. Under 'Automated Triggers', there is a section with a single checkbox: 'Disable automated triggers (only run manually)'.

Bff-with-trigger pipeline 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

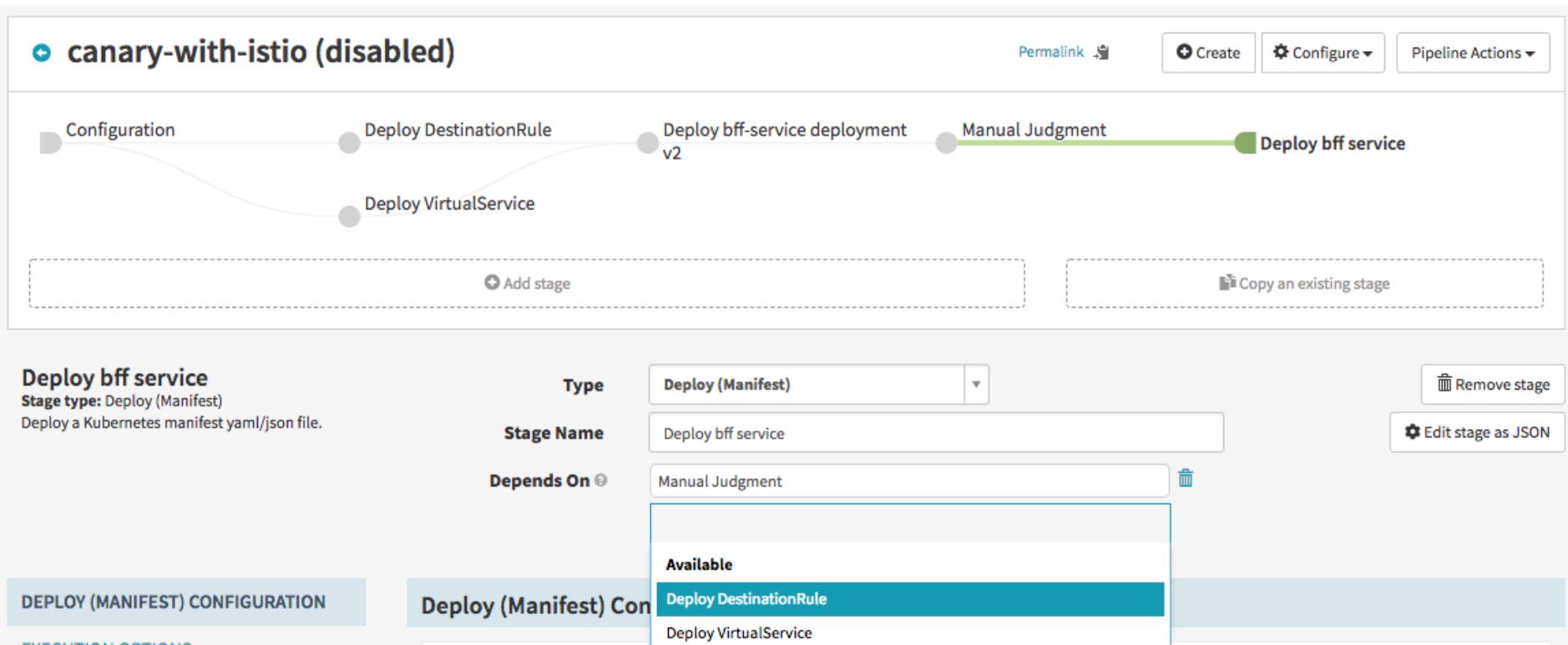
EXPECTED ARTIFACTS 해당 파이프라인이 실행될 때 산출될 것으로 기대되는 아티팩트 정의

Pipeline 생성

2. Stage 설정

Stage 기본 설정

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



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

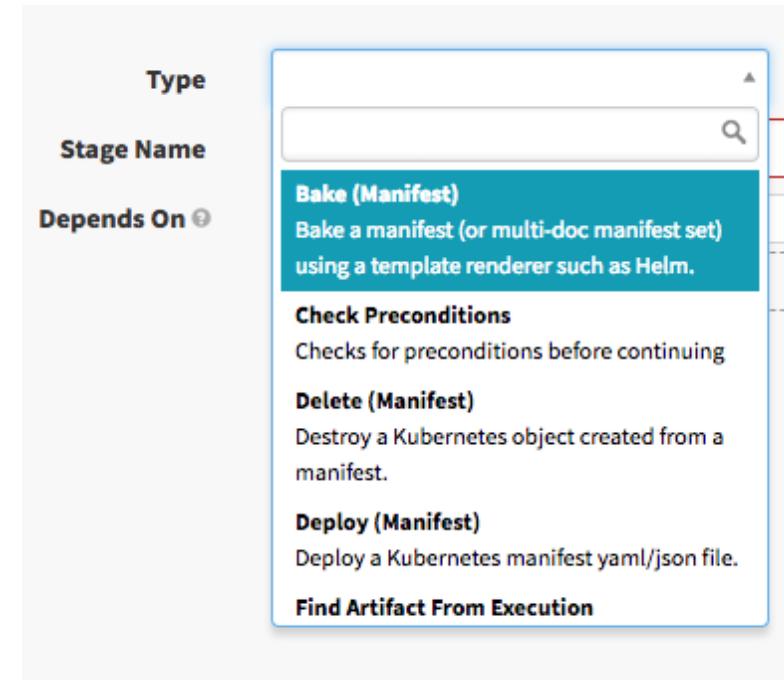
Scale Configuration K8S에 객체를 scaling

Script 스크립트 실행

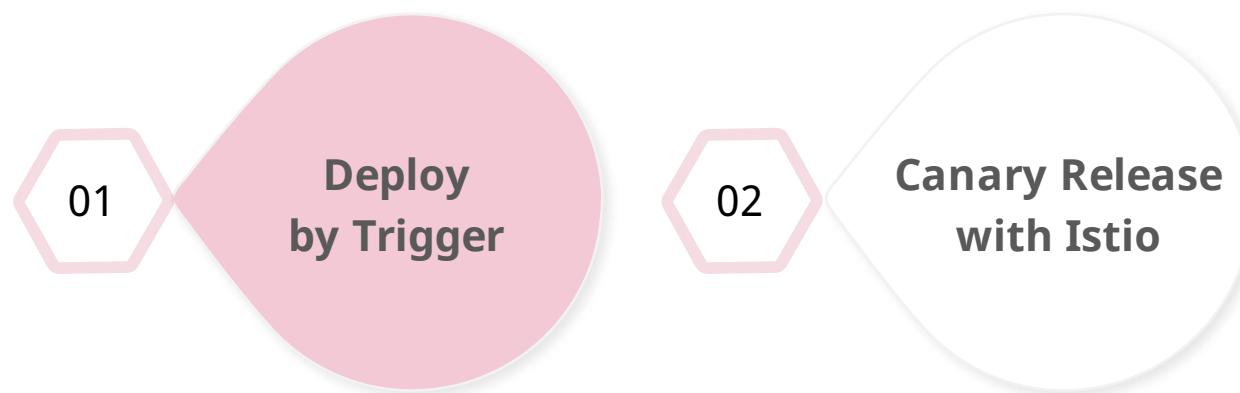
Undo Rollout K8S 객체의 상태를 원하는 revision number로 롤아웃

Wait 일정 시간 대기

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

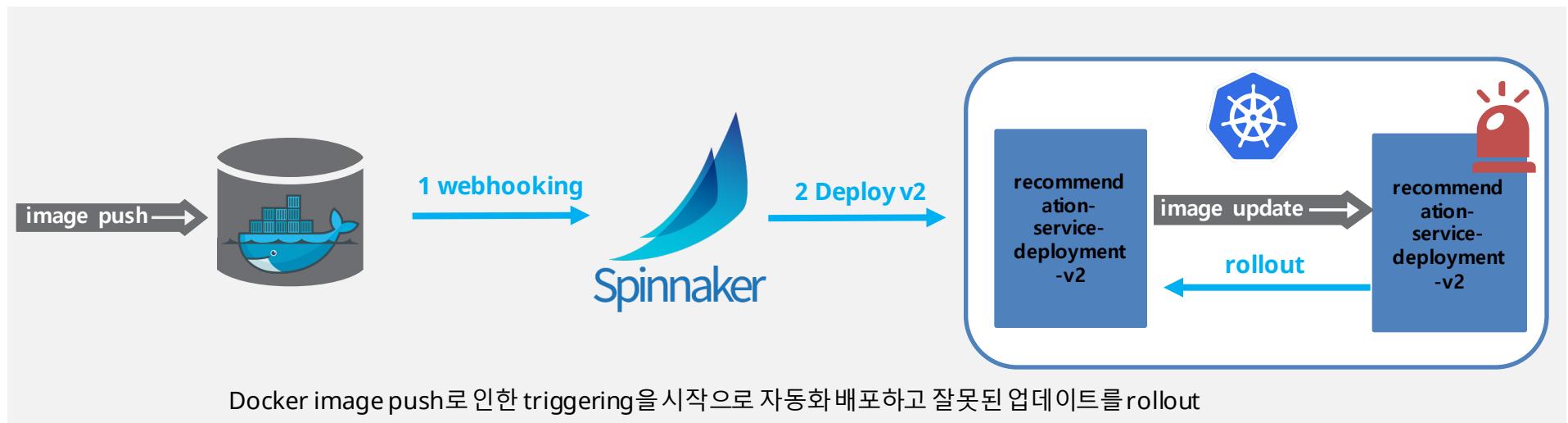


PART 02. Deploy Pipeline 구성하기



일반 배포 자동화

시나리오



일반 배포 자동화

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

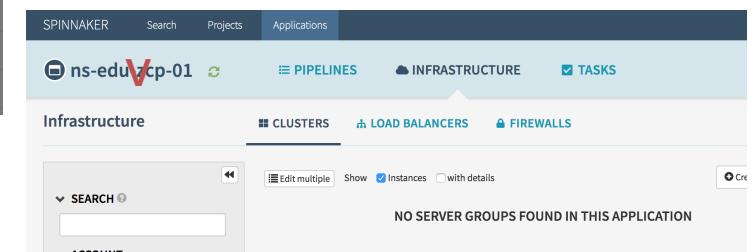


The screenshot shows the Spinnaker Dashboard's Applications page. At the top, there are tabs for SPINNAKER, Search, Projects, and Applications. The Applications tab is selected and has a red arrow pointing to it. On the right side of the page, there is a 'Actions' dropdown menu with a 'Create Application' option, also highlighted with a red arrow.

New Application

Name *	<input type="text" value="Enter an application name"/>	Labsflix\${number}
Owner Email *	<input type="text" value="Enter an email address"/>	
Repo Type	Select Repo Type	
Description	<input type="text" value="Enter a description"/>	
Instance Health		
<input type="checkbox"/> Consider only cloud provider health when executing tasks		
<input type="checkbox"/> Show health override option for each operation		
Instance Port	80	
Pipeline Behavior		
<input type="checkbox"/> Enable restarting running pipelines		
<small>* Required</small>		
<input type="button" value="Cancel"/> <input type="button" value="Create"/>		

✓ 생성된 Application 확인



The screenshot shows the Spinnaker Dashboard's Infrastructure page. At the top, there are tabs for SPINNAKER, Search, Projects, and Applications. The Applications tab is selected. In the center, there is a card for the application 'ns-eduVcp-01'. Below the card, there is a table with columns for CLUSTERS, LOAD BALANCERS, and FIREWALLS. A search bar is at the bottom left, and a message at the bottom right says 'NO SERVER GROUPS FOUND IN THIS APPLICATION'.

일반 배포 자동화

3 Pipeline 생성

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

No pipelines configured for this application.

[Configure a new pipeline](#)

3-2 Type: Pipeline 선택

Pipeline Name: 'Deploy recommendation v2' 입력
Create 버튼 클릭

Create New Pipeline

Type: Pipeline

Pipeline Name: Deploy recommendation v3

✓ Pipeline 생성 확인

✓ Pipeline 생성 확인

u-zcp-01 PIPELINES INFRASTRUCTURE TASKS

Deploy recommendation v3

Configuration

Concurrent Executions

Automated Triggers

일반 배포 자동화

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

Deploy recommendation v3

Configuration

CONCURRENT EXECUTIONS

AUTOMATED TRIGGERS (1)

PARAMETERS

NOTIFICATIONS

DESCRIPTION

EXPECTED ARTIFACTS

Automated Triggers

Type: Webhook Executes the pipeline when a webhook is received.
{ http://spinnaker-api.zcp-dtlabs.jp-tok.containers.mybluemix.net/webhooks/webhook/dockerhub-recommendation }

Source: dockerhub-recommendation

Payload Constraints

dtlabs/recommendation-service

PUBLIC REPOSITORY

Last pushed: an hour ago

Repo Info Tags Collaborators Webhooks Settings

Workflows

TRIGGER EVENT

WEB HOOKS +

Image Pushed

Spinnaker

When an image is pushed to this repo, your workflows will kick off based on your specified webhooks. [Learn More](#)

일반 배포 자동화

4 Parameter 설정

Namespace용 parameter를 생성한다.

Configuration stages의 Parameters에서 다음과 같이 설정

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

Parameters				
	Name	Label		
<input type="button" value="↑"/>	namespace		<input type="button" value="trash"/>	
Required	<input checked="" type="checkbox"/>			
Description	<input type="text"/>			
Default Value	<input type="text" value="본인의 namespace"/>			
Show Options	<input type="checkbox"/>			

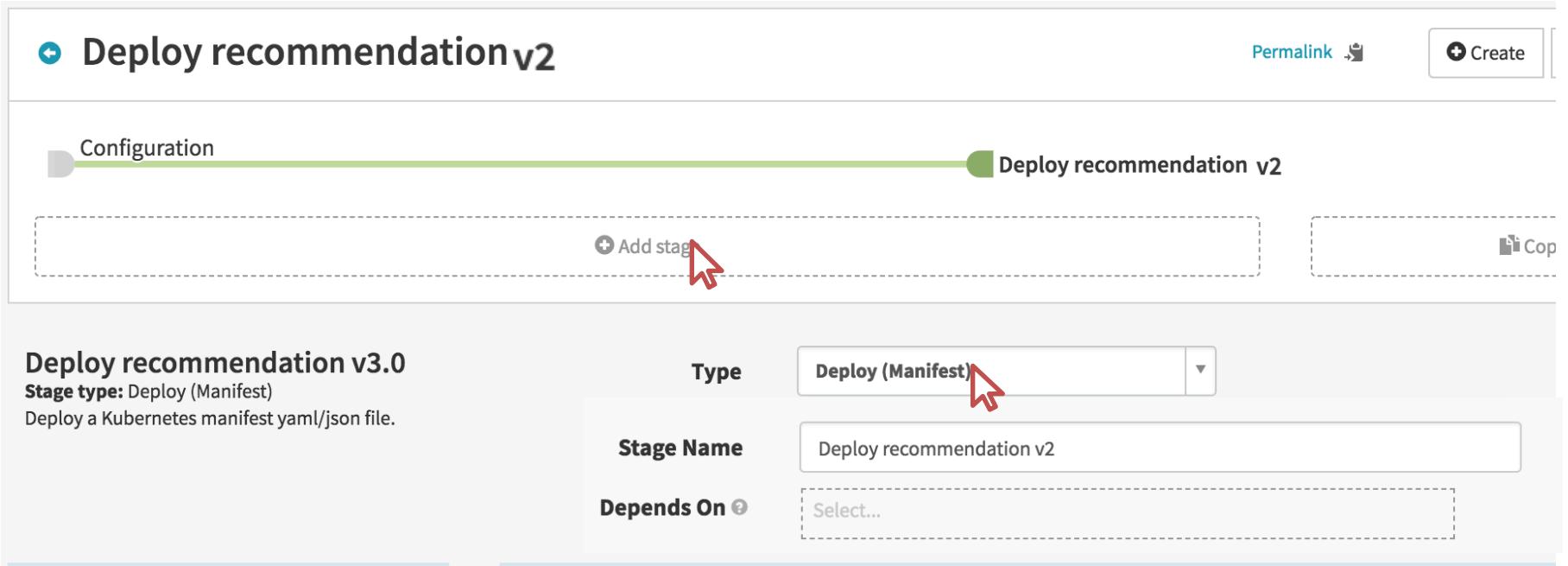
일반 배포 자동화

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



The screenshot shows a pipeline configuration interface for a deployment named "Deploy recommendation v2". A red arrow points to the "Add stage" button, which is highlighted with a red box. The configuration tab is selected. The pipeline consists of two stages: "Deploy recommendation v2" and "Deploy recommendation v3.0". The "Deploy recommendation v3.0" stage is set to "Deploy (Manifest)" type and depends on "Deploy recommendation v2".

Deploy recommendation v2

Configuration

Deploy recommendation v2

Add stage

Deploy recommendation v3.0

Stage type: Deploy (Manifest)

Deploy a Kubernetes manifest yaml/json file.

Type: Deploy (Manifest)

Stage Name: Deploy recommendation v2

Depends On: Select...

일반 배포 자동화

5 Kubernetes에 Deployment 배포

배포할 Deployment의 매니페스트를 입력한다.

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

Basic Settings

Account * Application *

Manifest Configuration

Manifest Source Text Artifacts

```
- apiVersion: apps/v1beta2
  kind: Deployment
  metadata:
    labels:
      app: recommendation-service
    version: v3
    name: recommendation-service-deployment-v3
    namespace: ns-zcp-edu-01
  spec:
    replicas: 1
    selector:
      matchLabels:
        app: recommendation-service
    template:
      metadata:
        labels:
          app: recommendation-service
      spec:
        containers:
          - env:
              - name: SPRING_PROFILES_ACTIVE
                value: k8s
            image: 'dtlabs/recommendation-service:3.0'
            imagePullPolicy: Always
            name: recommendation-service
            ports:
              - containerPort: 8080
```

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

```
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

일반 배포 자동화

5 Kubernetes에 Deployment 배포

- ✓ 생성된 파이프라인 확인

The screenshot shows a software interface for managing pipelines. At the top, there are tabs for 'Applications', 'Search', 'What's New', and 'Help'. Below the tabs, there are three main navigation buttons: 'PIPELINES' (with a red 'V' icon), 'INFRASTRUCTURE', and 'TASKS'. The 'PIPELINES' tab is selected. On the right side of the header, there is a 'CONFIG' button. The main content area displays a pipeline named 'Deploy recommendation V2'. This pipeline has a status bar indicating 'Trigger: disabled'. There are buttons for 'Create', 'Configure', and 'Start Manual Execution'. Below the pipeline name, there are filtering options: 'Group by Pipeline', 'Show 2 executions per pipeline', and 'stage durations'. A large, bold, blue button labeled 'Start Manual Execution' is prominently displayed at the bottom of the pipeline card.

6 Docker Image 푸시

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

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

일반 배포 자동화

6 Pipeline 실행 결과 확인

- ✓ Pipeline이 자동 시작하여 실행(RUNNING)

The screenshot shows a pipeline execution interface with the following details:

- Group by:** Pipeline
- Show:** 2 executions per pipeline
- Stage Durations:** Enabled
- Create**, **Configure**, and **Start Manual Execution** buttons

Deploy recommendation v2 (1) → 해당 pipeline에서 수행하고 있는 task 개수가 표시됨
Trigger: enabled

WEBHOOK a few seconds ago

Details (Mouse cursor is here)

Status: RUNNING Duration: 00:08

Deploy recommendation v3.0

STAGE DETAILS: DEPLOY RECOMMENDATION
Duration: 00:08

Step	Started	Duration	Status
Deploy recommendation v3.0	2018-08-23 09:49:33 PDT	00:08	RUNNING

DEPLOY RECOMMENDATION

Deploy Status	Task Status	Artifact Status
[Empty Box]		

Source | Permalink

일반 배포 자동화

6 Pipeline 실행 결과 확인

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

Deploy recommendation 1

Trigger: enabled [Configure](#) [▶ Start Manual Execution](#)

WEBHOOK
a few seconds ago
[▼ Details](#) 

Status: **RUNNING**  Duration: 00:46 

Deploy recommendation v3.0

STAGE DETAILS: DEPLOY RECOMMENDATION
Duration: 00:46

Step	Started	Duration	Status
Deploy recommendation v3.0	2018-08-23 09:49:33 PDT	00:46	RUNNING

▶ DEPLOY RECOMMENDATION 현재 수행하고 있는 Task 상태 실시간 모니터링

Deploy Status	Task Status	Artifact Status

Task	Duration
Deploy Manifest	00:00
Monitor Deploy	00:05
Promote Outputs	00:00
Force Cache Refresh	00:38
Wait For Manifest To Stabilize	-
Cleanup Artifacts	-
Bind Produced Artifacts	-

[Source](#) | [Permalink](#) 

일반 배포 자동화

6 Pipeline 실행 결과 확인

- ✓ Pipeline을 성공적으로 실행 완료되어 SUCCEEDED 표시된 결과 확인

Pipeline Execution Status: Status: SUCCEEDED

Deployment Details:

- Deployment: recommendation-service-deployment-v3 (available, stable)
- ScalingReplicaSet: 6 minutes ago, Scaled up replica set recommendation-service-deployment-v3-5ccf5976f5 to 1
- ScalingReplicaSet: 4 minutes ago, Scaled down replica set recommendation-service-deployment-v3-5ccf5976f5 to 0
- ScalingReplicaSet: 2 minutes ago, Scaled up replica set recommendation-service-deployment-v3-5ccf5976f5 to 1

Artifact Status:

- Consumed Artifacts: None
- Produced Artifacts: recommendation-service...

Summary: Pipeline 실행하여 생성된 Kubernetes 오브젝트

일반 배포 자동화

6 Pipeline 실행 결과 확인

- ✓ Infrastructure에서 Workloads 확인

The screenshot shows the CloudBees CI/CD Platform interface. The top navigation bar includes tabs for PIPELINES, INFRASTRUCTURE, and TASKS, with INFRASTRUCTURE selected. Below the navigation is a sub-navigation bar with CLUSTERS, LOAD BALANCERS, and FIREWALLS, with CLUSTERS selected. On the left, a sidebar provides filtering options for SEARCH, ACCOUNT, REGION, STACK, DETAIL, STATUS, and AVAILABILITY ZONES. The main content area displays a list of clusters under the heading NS-ZCP-EDU-01. One cluster, V001: dtlabs/recommendation-service:3.0, is shown with a green status icon and a progress bar indicating 1 ▲ : 100%. To the right, a detailed view of the 'recommendation-service-deployment' is shown. This view includes sections for INFORMATION (Created: 2018-08-23 09:49:34 PDT, Account: MY-K8S-V2-ACCOUNT, Namespace: ns-zcp-edu-01, Kind: deployment, Managing: replicaSet recommendation-service-deployment-v3-5ccf5976f5), STATUS (Available 2018-08-23T16:49:40Z, Progressing 2018-08-23T16:49:34Z), and EVENTS (1 × ScalingReplicaSet).

일반 배포 자동화

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

The screenshot shows two panels from the Spinnaker interface. The left panel displays a list of clusters, load balancers, and firewalls. It highlights the 'deployment recommendation-service-deployment' under the 'MY-K8S-V2-ACCOUNT' cluster. A specific pod, 'recommendation-service-deployment-v3-86c79fc4-xnngc', is selected, showing its status as 100% healthy. The right panel is a detailed view of the 'recommendation-service-deployment' server group. It includes sections for 'LABELS' (app: recommendation-service) and 'ANNOTATIONS'. The annotations list several key values:

```

artifact.spinnaker.io/location: ns-zcp-edu-01
artifact.spinnaker.io/name: recommendation-service-deployment-v3
artifact.spinnaker.io/type: kubernetes/deployment
deployment.kubernetes.io/revision: 2
moniker.spinnaker.io/application: ns-edu-zcp-01
moniker.spinnaker.io/cluster: deployment recommendation-service-deployment-v3

```

Below the annotations, there is a section for 'ARTIFACTS' which lists 'docker/image nginx'.

일반 배포 자동화

8 Rollout

최초 버전으로 rollout한다

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

The screenshot shows the deployment details for 'recommendation-service-deployment'. The deployment status is 1 ▲ : 100%. The dropdown menu for 'Server Group Manager Actions' is open, showing options like Scale, Undo Rollout, Edit, and Delete. A red arrow points to the 'Server Group Manager Actions' button.

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

The dialog box is titled 'Undo rollout of Deployment recommendation-service-deployment-v3 in ns-zcp-edu-01'. It has a 'Revision' dropdown set to 'Revision 1 (replicaSet recommendation-service-deployment-v3-5ccf6976f5)' and a 'Reason' text area. At the bottom are 'Cancel' and 'Submit' buttons, with a red arrow pointing to the 'Submit' button.

일반 배포 자동화

8 Rollout

Undo rollout of deployment recommendation-service-deployment-v3 in ns-zcp-edu-01

- ✓ Undo Rollout Manifest (00:00)
- ✓ Monitor Undo Rollout (00:05)
- ✓ Force Cache Refresh (00:01)
- ✓ Wait For Stable (00:57)

Operation succeeded!

You can [monitor this task from the Tasks view.](#)

Rollout 실행 완료함

nginx가 적용된 pod가 없어지고 예전 Image가 적용된 pod가 생성됨을 확인

ns-edu-zcp-01

Pipelines Infrastructure Tasks

CONFIG

Infrastructure

CLUSTERS LOAD BALANCERS FIREWALLS

SEARCH

ACCOUNT

- my-k8s-v2-account

REGION

- ns-zcp-edu-01

STACK

- (none)

DETAIL

- (none)

STATUS

MY-K8S-V2-ACCOUNT deployment recommendation-service-deployment NS-ZCP-EDU-01

V003: dtlabs/recommendation-service:3.0 100%

V002: nginx 0%

recommendation-service-deployment

Server Group Manager Actions

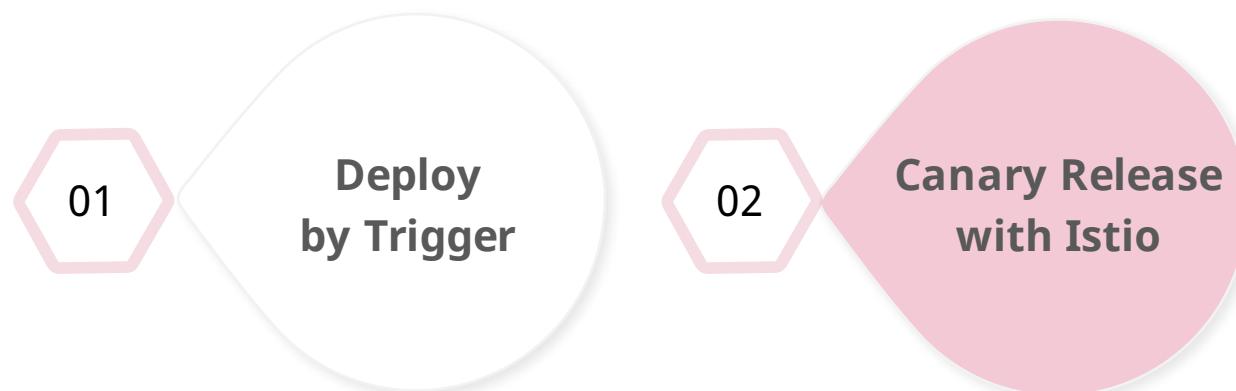
Managing service-deployment-v3-86c7f9fc4

replicaSet recommendation-service-deployment-v3-5ccf5976f5

STATUS

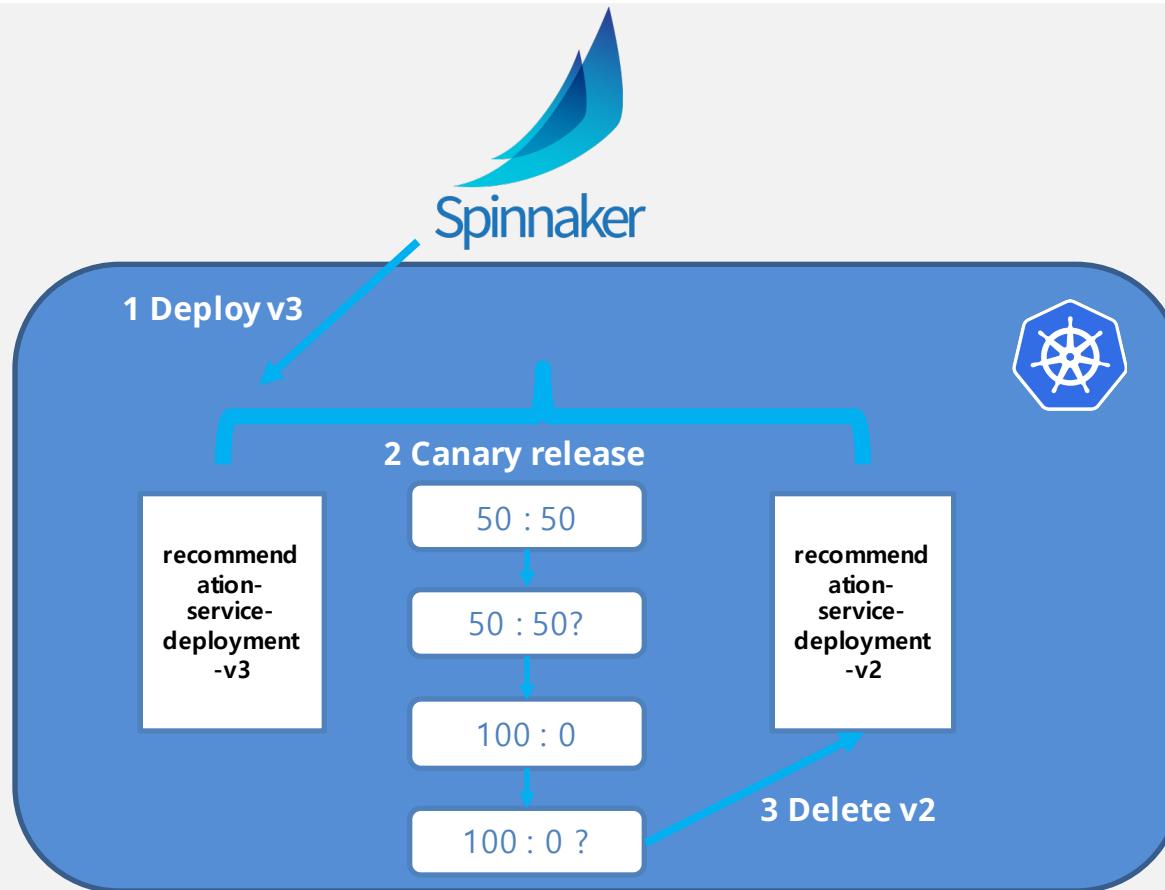
Available 2018-08-23T16:49:40Z Deployment has minimum availability.

PART 02. Deploy Pipeline 구성하기



Istio를 활용한 Canary 배포

시나리오

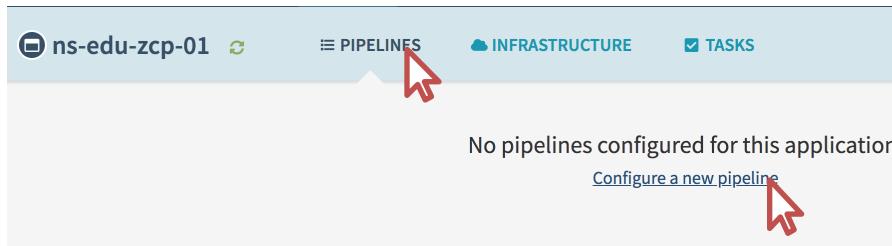


새 버전 애플리케이션을 canary release하고 기존 버전 삭제 (Parameter 사용)

Istio를 활용한 Canary 배포

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 Parameters에서 3개를 다음과 같이 설정

- **Name:** namespace
- **Name:** app_old_version
- **Name:** app_new_version

Parameters		
<input type="text"/>	Name	namespace
<input checked="" type="checkbox"/>	Required	<input checked="" type="checkbox"/>
<input type="text"/>	Description	namespace 전체
<input type="text"/>	Default Value	
<input type="checkbox"/>	Show Options	
<input type="text"/>	Name	app_old_version
<input checked="" type="checkbox"/>	Required	<input checked="" type="checkbox"/>
<input type="text"/>	Description	기존 버전. ex.v2
<input type="text"/>	Default Value	v2
<input type="checkbox"/>	Show Options	
<input type="text"/>	Name	app_new_version
<input checked="" type="checkbox"/>	Required	<input checked="" type="checkbox"/>
<input type="text"/>	Description	label 값, deployment 이름 뒤에 붙는 버전 ex.v3
<input type="text"/>	Default Value	v3

Istio를 활용한 Canary 배포

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

The screenshot shows the 'Deploy (Manifest) Configuration' screen. On the left, there's a sidebar with tabs: 'DEPLOY (MANIFEST) CONFIGURATION' (selected), 'EXECUTION OPTIONS', 'NOTIFICATIONS', 'COMMENTS', and 'PRODUCES ARTIFACTS'. The main area has two sections: 'Basic Settings' and 'Manifest Configuration'. In 'Basic Settings', 'Account' is set to 'my-k8s-v2-account' and 'Application' is set to 'ns-edu-zcp-01'. In 'Manifest Configuration', 'Manifest Source' is set to 'Text', indicated by a red arrow pointing to the 'Text' radio button. Below it is a code editor containing a Deployment manifest for 'recommendation-service'. A large red arrow points from this code editor area towards the right side of the slide, where the manifest is shown in its entirety.

```

apiVersion: apps/v1beta2
kind: Deployment
metadata:
  labels:
    app: recommendation-service
  name: recommendation-service-deployment-v3
  namespace: ns-zcp-edu-01
spec:
  replicas: 1
  selector:
    matchLabels:
      app: recommendation-service
  template:
    metadata:
      labels:
        app: recommendation-service
    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

```

```

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

```

Istio를 활용한 Canary 배포

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



```

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

```

5 Manual Judgment

사용자가 수동으로 5:5로 라우팅이 되는지 확인하는 stage를 추가

Add stage 클릭 - **Stage Type: ManualJudgement**선택

Stage Name: 5:5?

Depends On: Deploy 5:5 선택

Istio를 활용한 Canary 배포

6 istio rounting 10:0 적용

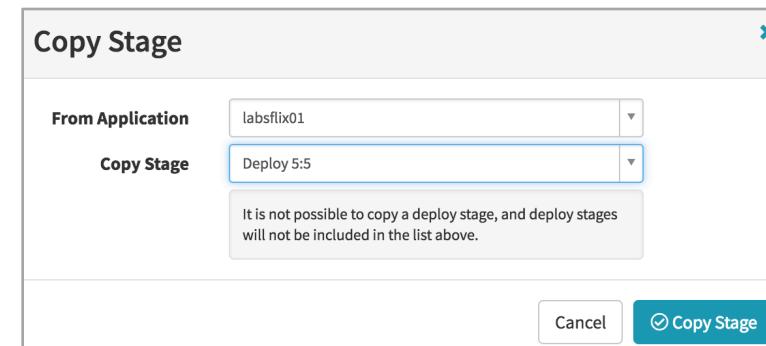
10:0 비율로 라우팅하기 위해 istio 오브젝트 배포한다.
이전에 만든 'Deploy 5:5' stages를 복사하여 사용한다.

6-1 Copy an existing stage 클릭

The screenshot shows the 'Canary release' pipeline configuration. At the top, there's a horizontal timeline with stages: Configuration, Deploy recommendation - new, Deploy 5:5, 5:5?, and Deploy 10:0. Below the timeline, the Deploy 10:0 stage is selected. On the left, there's a note about deploying a Kubernetes manifest. On the right, there are fields for 'Stage Name' (Deploy 10:0) and 'Depends On' (5:5?). Buttons for 'Add stage', 'Remove stage', and 'Edit stage as JSON' are also present. The 'Copy an existing stage' button is highlighted with a red arrow.

6-2 Copy Stage 팝업창에서 다음과 같이 선택

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



Istio를 활용한 Canary 배포

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 되는지 확인하는 stage를 추가

Add stage 클릭 - **Stage Type: ManualJudgement** 선택

Stage Name: 10:0?

Depends On: Deploy 10:0 선택

Istio를 활용한 Canary 배포

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}

Delete (Manifest) Configuration

Warning! This stage is under active development and is subject to change.

Manifest

Account	my-k8s-v2-account
Namespace	\${parameters.namespace}
Match On	<input type="radio"/> Labels <input checked="" type="radio"/> Name
Kind	deployment
Name	recommendation-service-deployment-\${parameters.app_old_version}

Istio를 활용한 Canary 배포

9 Pipeline 실행 결과 확인

1 Pipeline 화면에서 Canary release pipeline 수동 실행

The screenshot shows the Pipeline interface with the following details:

- Pipelines:** Shows 2 executions per pipeline.
- INFRASTRUCTURE:** Selected tab.
- TASKS:** Checked checkbox.
- Actions:** Create, Configure, Start Manual Execution.
- Pipeline Details:** Name: Canary release, Trigger: disabled, Configuration, Start Manual Execution.

2 Confirm Execution 설정

- **namespace:** 본인의 namespace명 전체 입력

3 Run 버튼 클릭

The dialog box contains the following information:

This will start a new run of **Canary release**.

This pipeline is parameterized. Please enter values for the parameters below.

namespace *	<input type="text"/>
app_old_version *	v2
app_new_version *	v3

* Required

Notifications Notify me when the pipeline completes

Cancel **Run**

감사합니다

