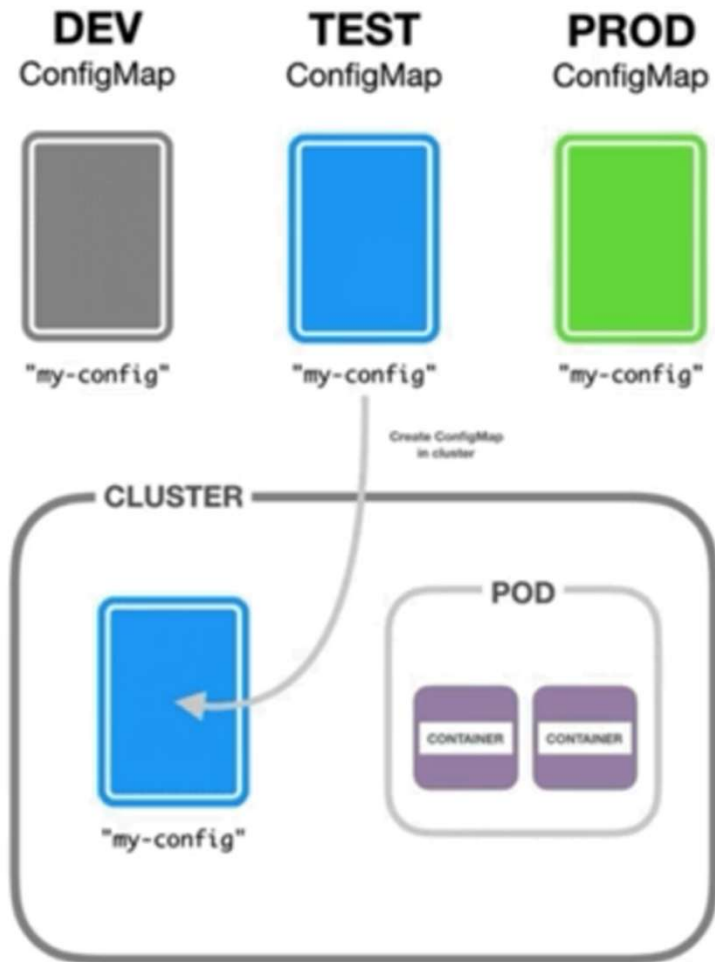


ConfigMap




ConfigMap 리소스

- 쿠버네티스에서 비밀정보가 아닌 설정 데이터를 키-값(key-value) 형태로 저장하는 오브젝트
- 애플리케이션의 환경 변수, 명령줄 인수, 설정 파일 등을 컨테이너와 분리해 관리할 때 사용하며, 설정 변경 시 애플리케이션 재배포 없이 유연하게 업데이트할 수 있음.

주요 특징

- 평문으로 관리
 - 패스워드, API 키 등 민감한 데이터는 Secret 오브젝트로 관리해야 함.
 - ConfigMap 데이터는 Base64 인코딩 없이 일반 텍스트로 저장됨
- 데이터 형태
 - 키=값 쌍 (e.g., LOG_LEVEL=DEBUG).
 - 전체 설정 파일 (e.g., application.properties, nginx.conf).
- 사용 시점
 - 파드 생성 시 데이터를 주입하거나, 런타임 중에 볼륨으로 마운트해 사용함.
- 네임스페이스 범위
 - ConfigMap은 특정 네임스페이스에 속하며, 동일 네임스페이스의 파드만 참조 가능


ConfigMap 리소스 적용 방식



```
+  
(ex. , API
```

- ConfigMap의 값을 컨테이너의 환경변수로 사용 (데이터 주입)

- ConfigMap의 값을 파드 볼륨으로 마운트하여 사용 (파일 읽어 오기)

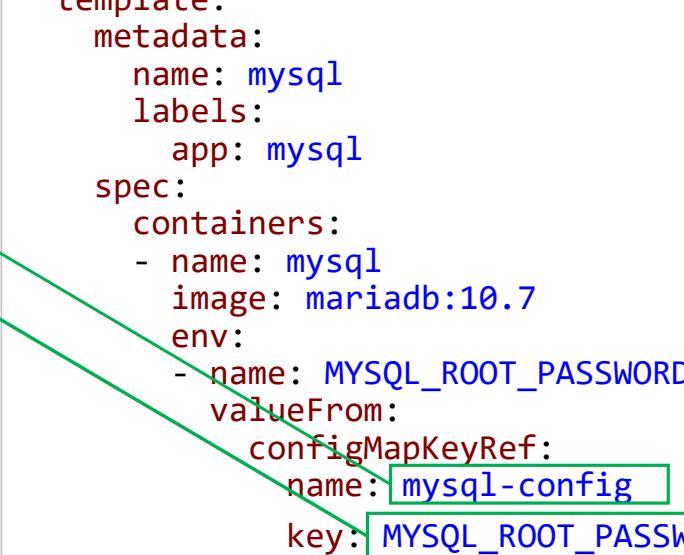


```
+ 가  
+  
(ex. Nginx , Apache  
Spring Boot application.yaml )
```

ConfigMap 리소스 적용 방식 - 환경변수

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: mysql-config
data:
  MYSQL_ROOT_PASSWORD: password123
  MYSQL_DATABASE: order
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql
spec:
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      name: mysql
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mariadb:10.7
          env:
            - name: MYSQL_ROOT_PASSWORD
              valueFrom:
                configMapKeyRef:
                  name: mysql-config
                  key: MYSQL_ROOT_PASSWORD
```



ConfigMap 리소스 적용 방식 - 볼륨 마운트

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: nginx-config
data:
  # 키=파일명, 값=파일내용
  nginx.conf: |
    server {
      listen 80;
      server_name localhost;

      location / {
        root /usr/share/nginx/html;
        index index.html;
        # 특별 설정: 캐시 비활성화
        add_header Cache-Control "no-store";
      }

      error_page 500 502 503 504 /50x.html;
    }
```

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
    - name: nginx-container
      image: nginx:1.25
      volumeMounts:
        - name: nginx-config-volume # 볼륨 이름
          mountPath: /etc/nginx/conf.d # 마운트 경로
          readOnly: true # 읽기 전용
      volumes:
        - name: nginx-config-volume # 볼륨 정의
          configMap:
            name: nginx-config # 사용할 ConfigMap
            items:
              - key: nginx.conf # ConfigMap 내 키
                path: nginx.conf
```

컨테이너 내 경로 = mountPath + items.path

/etc/nginx/conf.d/nginx.conf

해당 볼륨 (/etc/nginx/conf.d) 의 지정된 파일 (nginx.conf) 에 데이터 작성

ConfigMap 실습

환경변수 방식

backend user : ConfigMap 작성 및 등록

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: backend-user-config
data:
  server.port: "8081"
```

입력을 통해 생성 파일을 통해 생성 서식을 통해 생성

현재 선택된 네임스페이스에 생성할 리소스를 명시하는 YAML

```
1 apiVersion: v1
2 kind: ConfigMap
3 metadata:
4   name: backend-user-config
5 data:
6   server.port: "8081"
```

업로드 Cancel

backend user : Deployment 수정

YAML

JSON

```
113 creationTimestamp: null
114 labels:
115   app: k8s-backend-user
116 annotations:
117   kubernetes.io/restartedAt: '2025-06-10T12:48:51Z'
118 spec:
119   containers:
120   - name: k8s-backend-user
121     image: solarhc/k8s-backend-user:0.2.3
122     ports:
123     - containerPort: 8080
124       protocol: TCP
125     env:
126     - name: SPRING_PROFILES_ACTIVE
127       value: dev
128     - name: server.port
129       valueFrom:
130         configMapKeyRef:
131           name: backend-user-config
132           key: server.port
133   resources: {}
134   terminationMessagePath: /dev/termination-log
135   terminationMessagePolicy: File
```

이 액션은 다음 커맨드와 동일합니다. `kubectl apply -f <spec.yaml>`

Update

Cancel

Deployments

이름	이미지	레이블	파트	생성 시간	
k8s-backend-alim-deployment	solarhc/k8s-backend-alim:0.2.3	-	1 / 1	a.day.ago	⋮
nd-user-deployment	solarhc/k8s-backend-user:0.2.3	-	1 / 1	a.day.ago	⋮
teaway-deployment	solarhc/k8s-api-gateway:0.0.5	-	1 / 1	a.day.ago	⋮

Scale

Edit

Restart

Delete

...

template:

metadata:

labels:

app: k8s-backend-user

spec:

containers:

- name: k8s-backend-user

image: solarhc/k8s-backend-user:0.0.1

imagePullPolicy: Always

env:

- name: SPRING_PROFILES_ACTIVE

value: dev

- name: server.port

valueFrom:

configMapKeyRef:

name: backend-user-config

key: server.port

ports:

- containerPort: 8080

...

ConfigMap 실습

볼륨 마운트 방식

backend user : 기존 ConfigMap 제거

Config Maps			
이름	레이블	생성 시간 ↑	
backend-user-config	-	16 minutes ago	⋮
kube-root-ca.crt	-	a day ago	


Edit

Delete

backend user : 볼륨 마운트 방식 ConfigMap 작성 및 등록

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: backend-user-config
data:
  applicatoin-dev.yml: |
    server:
      port: 8080

    spring:
      datasource:
        url: jdbc:mysql://k8s-external-user-mysql-
service:3306/user?serverTimezone=UTC&useSSL=true&autoReconnect=tru
e&useUnicode=true&characterEncoding=utf-8
        username: user
        password: 1234
        driver-class-name: com.mysql.cj.jdbc.Driver
      hikari:
        connection-test-query: SELECT 1 # HikariCP 유효성 검사 추가
        validation-timeout: 5000
    jpa:
      hibernate:
        ddl-auto: create # 오직 테스트 환경에서만
        generate-ddl: true # 오직 테스트 환경에서만
(spring.jpa.generate-ddl)
      show-sql: true
      open-in-view: false
    kafka:
      bootstrap-servers: 192.168.0.135:9092
```



입력을 통해 생성 파일을 통해 생성 서식을 통해 생성

현재 선택된 네임스페이스에 생성할 리소스를 명시하는 YAML 또는 JSON

```
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: backend-user-config
5  data:
6    applicatoin-dev.yml: |
7      server:
8        port: 8080
9
10     spring:
11       datasource:
12         url: jdbc:mysql://k8s-external-user-mysql-service:330
           =utf-8
13         username: user
14         password: 1234
15         driver-class-name: com.mysql.cj.jdbc.Driver
16       hikari:
17         connection-test-query: SELECT 1 # HikariCP 유효성 검
18         validation-timeout: 5000
19     jpa:
20       hibernate:
21         ddl-auto: create # 오직 테스트 환경에서만
22         generate-ddl: true # 오직 테스트 환경에서만 (spring.i
```

업로드 Cancel

backend user : application-dev.yml 수정

```
spring:  
  config:  
    import: file:/etc/config/application-dev.yml
```

applicatoin-dev.yml

backend user : Deployment 수정

containers:

- name: k8s-backend-user
- image: solarhc/k8s-backend-user:0.0.1
- imagePullPolicy: Always
- env:
 - name: SPRING_PROFILES_ACTIVE
 - value: dev
- ports:
 - containerPort: 8080

volumeMounts:

- mountPath: /etc/config
- name: backend-user-config-volume

volumes:

- name: backend-user-config-volume
- configMap:
 - name: backend-user-config
 - items:
 - key: application-dev.yml
 - path: application-dev.yml

Edit a resource

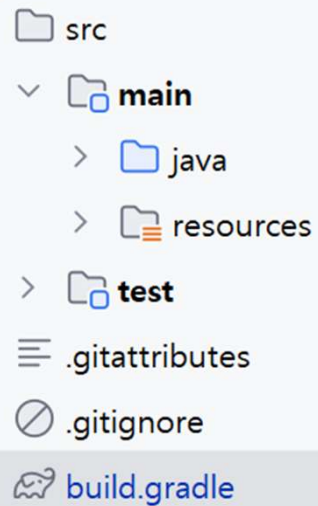
YAML	JSON
121 -	image: solarhc/k8s-backend-user:0.0.1
122 -	ports:
123 -	- containerPort: 8080
124 -	protocol: TCP
125 -	env:
126 -	- name: SPRING_PROFILES_ACTIVE
127 -	value: dev
128 -	volumeMounts:
129 -	- mountPath: /etc/config
130 -	name: backend-user-config-volume
131 -	resources: {}
132 -	terminationMessagePath: /dev/termination-log
133 -	terminationMessagePolicy: File
134 -	imagePullPolicy: Always
135 -	volumes:
136 -	- name: backend-user-config-volume
137 -	configMap:
138 -	name: backend-user-config
139 -	items:
140 -	- key: application-dev.yml
141 -	path: application-dev.yml
142 -	restartPolicy: Always
143 -	terminationGracePeriodSeconds: 30

이 액션은 다음 커맨드와 동일합니다. `kubectl apply -f <spec.yaml>`

[Update](#) [Cancel](#)

(key = ConfigMap의 키 값 & path = 어플리케이션에 적용될 파일 경로)

backend-user : 앱 버전 업그레이드



A file explorer view showing the project structure. The 'src' directory is expanded, showing 'main' and 'test' subdirectories. 'main' contains 'java' and 'resources' subdirectories. The root directory contains '.gitattributes', '.gitignore', and 'build.gradle'.

- src
 - main
 - java
 - resources
 - test
- .gitattributes
- .gitignore
- build.gradle

```
plugins {  
    id 'java'  
    id 'org.springframework.boot' version '3.5.0'  
    id 'io.spring.dependency-management' version '1.1.7'  
}  
  
group = 'com.welab'  
version = '0.0.5'  
  
...
```

backend-user : 코드 commit & push

Commit & Push to Github

backend-user : Jenkins Pipeline 빌드

Status

Changes

▶ 파라미터와 함께 빌드

구성

Pipeline 삭제

Move

Full Stage View

Stages

Rename

Pipeline Syntax

Pipeline kube-backend-user

매개변수가 필요한 빌드입니다.

TAG

origin/main



☐ RELEASE

▶ 매개변수가 필요한 빌드입니다.

Cancel

Declarative: Checkout SCM	Declarative: Tool Install	Set Version	Build & Test Application	Build Docker Image	Push Docker Image
4s	367ms	1min 14s	49s	7s	11s
4s	367ms	1min 14s	49s	7s	11s

backend-user : Deployment 이미지 버전 업데이트

Deployments					
이름	이미지	레이블	파드	생성 시간 ↑	
 k8s-backend-alim-deployment	solarhc/k8s-backend-alim:0.0.1	-	1 / 1	16 minutes ago	⋮
 k8s-backend-user-deployment	solarhc/k8s-backend-user:0.0.1	-	1 / 1	2 hours ago	⋮

Scale

Edit

Restart

Delete

```
128 spec:
129   volumes:
130     - name: backend-user-config-volume
131       configMap:
132         name: backend-user-config
133         items:
134           - key: application-dev.yml
135             path: application-dev.yml
136       defaultMode: 420
137   containers:
138     - name: k8s-backend-user
139       image: namsangwon/k8s-backend-user:0.0.7
140       ports:
141         - containerPort: 8080
142           protocol: TCP
143       env:
144         - name: SPRING_PROFILES_ACTIVE
145           value: dev
146       resources: {}
147       volumeMounts:
148         - name: backend-user-config-volume
149           mountPath: /etc/config
```

볼륨 마운트 방식

ConfigMap 실습

api-gateway

api-gateway : 볼륨 마운트 방식 ConfigMap 작성 및 등록

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: api-gateway-config
data:
  application-dev.yml: |
    server:
      port: 8080

    spring:
      cloud:
        gateway:
          mvc:
            routes:
              - id: backend-user
                uri: http://k8s-backend-user-service:8080
                predicates:
                  - Path=/api/user/**
              - id: backend-alim
                uri: http://k8s-backend-alim-service:8080
                predicates:
                  - Path=/api/alim/**
              - id: backend-post
                uri: http://k8s-backend-post-service:8080
                predicates:
                  - Path=/api/post/**
```

입력을 통해 생성

파일을 통해 생성

서식을 통해 생성

현재 선택된 네임스페이스에 생성할 리소스를 명시하는 YAML 또는 JSON

```
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: api-gateway-config
5  data:
6    application-dev.yml: |
7      server:
8        port: 8080
9
10     spring:
11       cloud:
12         gateway:
13           mvc:
14             routes:
15               - id: backend-user
16                 uri: http://k8s-backend-user-service:8080
17                 predicates:
18                   - Path=/api/user/**
19               - id: backend-alim
20                 uri: http://k8s-backend-alim-service:8080
21                 predicates:
```

업로드

Cancel

api-gateway : application-dev.yml 수정

```
spring:  
  config:  
    import: file:/etc/config/application-dev.yml
```

application-dev.yml

api-gateway : Deployment 수정

containers:

- **name:** k8s-api-gateway
- image:** solarhc/k8s-api-gateway:0.0.1
- imagePullPolicy:** Always

env:

- **name:** SPRING_PROFILES_ACTIVE
- value:** dev

ports:

- **containerPort:** 8080

volumeMounts:

- **mountPath:** /etc/config
- name:** api-gateway-config-volume

volumes:

- **name:** api-gateway-config-volume
- configMap:**
- name:** api-gateway-config
- items:**
- **key:** application-dev.yml
- path:** application-dev.yml

Edit a resource

YAML	JSON
113	spec:
114	replicas: 1
115	selector:
116	matchLabels:
117	app: k8s-api-gateway
118	template:
119	metadata:
120	creationTimestamp: null
121	labels:
122	app: k8s-api-gateway
123	spec:
124	volumes:
125	- name: api-gateway-config-volume
126	configMap:
127	name: api-gateway-config
128	items:
129	- key: application-dev.yml
130	path: application-dev.yml
131	defaultMode: 420
132	containers:
133	- name: k8s-api-gateway
134	image: solarhc/k8s-api-gateway:0.0.1

 이 액션은 다음 커맨드와 동일합니다. `kubectl apply -f <spec.yaml>`

Update

Cancel

api-gateway : 앱 버전 업그레이드

```
plugins {  
    id 'java'  
    id 'org.springframework.boot' version '3.5.0'  
    id 'io.spring.dependency-management' version '1.1.7'  
}  
  
group = 'com.welab'  
version = '0.0.3'  
  
...
```

build.gradle

api-gateway : 코드 commit & push

Commit & Push to Github

api-gateway : Jenkins Pipeline 빌드

Status

Changes

▶ 파라미터와 함께 빌드

구성

Pipeline 삭제

Move

Full Stage View

Stages

Rename

Pipeline Syntax

Pipeline kube-api-gateway

매개변수가 필요한 빌드입니다.

TAG

origin/main

☐ RELEASE

▶ 매개변수가 필요한 빌드

Declarative: Checkout SCM	Declarative: Tool Install	Set Version	Build & Test Application	Build Docker Image	Push Docker Image
2s	385ms	32s	59s	0ms	0ms
2s	385ms	32s	59s		

api-gateway : Deployment 이미지 버전 업데이트

Deployments				
이름	이미지	레이블	파드	생성 시간 ↑
k8s-backend-alim-deployment	solarhc/k8s-backend-alim:0.0.1	-	1 / 1	20.minutes.ago
k8s-backend-user-deployment	solarhc/k8s-backend-user:0.0.2	-	1 / 1	2.hours.ago
k8s-api-gateway-deployment	solarhc/k8s-api-gateway:0.0.3	-	1 / 1	2.hours.ago

Scale

Edit

Restart

Delete

```
97 spec:
98   replicas: 1
99   selector:
100     matchLabels:
101       app: k8s-api-gateway
102   template:
103     metadata:
104       creationTimestamp: null
105     labels:
106       app: k8s-api-gateway
107   spec:
108     containers:
109     - name: k8s-api-gateway
110       image: solarhc/k8s-api-gateway:0.0.2
111     ports:
112     - containerPort: 8080
113       protocol: TCP
114     env:
```

i 이 액션은 다음 커맨드와 동일합니다. `kubectl apply -f <spec.yaml>`

Update

Cancel

볼륨 마운트 방식

ConfigMap 실습

backend-alim

backend-alim : 볼륨 마운트 방식 ConfigMap 작성 및 등록

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: backend-alim-config
data:
  application-dev.yml: |
    server:
      port: 8080

    spring:
      kafka:
        bootstrap-servers: k8s-external-kafka-service:9092
```

입력을 통해 생성 파일을 통해 생성 서식을 통해 생성

현재 선택된 네임스페이스에 생성할 리소스를 명시하는 YAML 또는 JSON

```
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: backend-alim-config
5  data:
6    application-dev.yml: |
7      server:
8        port: 8080
9
10     spring:
11       kafka:
12         bootstrap-servers: k8s-external-kafka-service:9092
```

업로드 Cancel

backend-alim : application-dev.yml 수정

```
spring:  
  config:  
    import: file:/etc/config/application-dev.yml
```

application-dev.yml

backend-alim : Deployment 수정

containers:

- **name:** k8s-backend-alim
image: solarhc/k8s-backend-alim:0.0.1
imagePullPolicy: Always

env:

- **name:** SPRING_PROFILES_ACTIVE
value: dev

ports:

- **containerPort:** 8080

volumeMounts:

- **mountPath:** /etc/config
name: backend-alim-config-volume

volumes:

- **name:** backend-alim-config-volume
configMap:
name: backend-alim-config
items:
- **key:** application-dev.yml
path: application-dev.yml

Edit a resource

YAML JSON

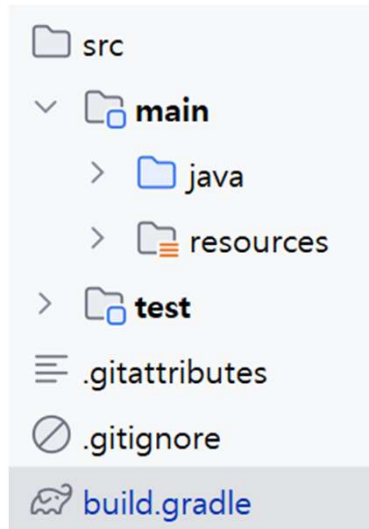
```
112 - containerPort: 8080
113   protocol: TCP
114
115 env:
116   - name: SPRING_PROFILES_ACTIVE
117     value: dev
118
119 volumeMounts:
120   - mountPath: /etc/config
121     name: backend-alim-config-volume
122
123 resources: {}
124
125 terminationMessagePath: /dev/termination-log
126
127 terminationMessagePolicy: File
128
129 imagePullPolicy: Always
130
131 volumes:
132   - name: backend-alim-config-volume
133     configMap:
134       name: backend-alim-config
135       items:
136         - key: application-dev.yml
137           path: application-dev.yml
138
139 restartPolicy: Always
140
141 terminationGracePeriodSeconds: 30
```

 이 액션은 다음 커맨드와 동일합니다. `kubectl apply -f <spec.yaml>`

Update

Cancel

backend-alim : 앱 버전 업그레이드



```
plugins {  
    id 'java'  
    id 'org.springframework.boot' version '3.5.0'  
    id 'io.spring.dependency-management' version '1.1.7'  
}  
  
group = 'com.welab'  
version = '0.0.3'  
  
...
```

backend-alim : 코드 commit & push

Commit & Push to Github

backend-alim : Jenkins Pipeline 빌드

Status

Changes

▶ 파라미터와 함께 빌드

구성

Pipeline 삭제

Move

Full Stage View

Stages

Rename

Pipeline Syntax

Pipeline kube-backend-alim

매개변수가 필요한 빌드입니다.

TAG

origin/main




☐ RELEASE

▶ 매개변수가 필요한 빌드입니다.

Cancel


Declarative: Checkout SCM	Declarative: Tool Install	Set Version	Build & Test Application	Build Docker Image	Push Docker Image
4s	367ms	1min 14s	49s	7s	11s
4s	367ms	1min 14s	49s	7s	11s

backend-alim : Deployment 이미지 버전 업데이트

Deployments					
이름	이미지	레이블	파드	생성 시간 ↑	
 k8s-backend-alim-deployment	solarhc/k8s-backend-alim:0.0.1	-	1 / 1	16 minutes ago	⋮
 k8s-backend-user-deployment	solarhc/k8s-backend-user:0.0.1	-	1 / 1	2 hours ago	⋮
 k8s-api-gateway-deployment	solarhc/k8s-api-gateway:0.0.3	-	1 / 1	2 hours	⋮

- Scale
- Edit
- Restart
- Delete

```
97 spec:
98   replicas: 1
99   selector:
100     matchLabels:
101       app: k8s-backend-alim
102   template:
103     metadata:
104       creationTimestamp: null
105     labels:
106       app: k8s-backend-alim
107   spec:
108     containers:
109       - name: k8s-backend-alim
110         image: solarhc/k8s-backend-alim:0.0.2
111         ports:
```

 이 액션은 다음 커맨드와 동일합니다. `kubectl apply -f <spec.yaml>`

[Update](#) [Cancel](#)