

Secret

Secret 리소스

- 민감한 정보(비밀번호, API 키, 인증서 등)를 안전하게 저장하고 관리하기 위한 오브젝트
- ConfigMap과 유사하지만 보안이 강화된 방식으로 데이터를 처리

주요 특징

- 암호화 저장
 - 기본적으로 Base64 인코딩으로 저장
 - etcd 암호화 활성화 시 데이터베이스 레벨에서 암호화 저장 가능
- 접근 제어
 - RBAC를 통한 세분화된 권한 관리

Secret 작성 방법

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-user-secret
type: Opaque
stringData: # 평문으로 직접 입력
             # (자동 Base64 변환)
username: "admin"
password: "P@ssw0rd!123"
config.yaml: | # 멀티라인 데이터도 지원
  database:
    host: db.example.com
    port: 5432
    ssl: true
```

Secret 작성

```
kind: Secret
apiVersion: v1
metadata:
  name: test-backend-user-secret
data:
  username: YWRtaW4=
  password: UEBzc3cwcmQhMTIz
  config.yaml: >-
    ZGF0YUJhc2U6CiAgaG9zdDogZGIuZXhhbXBsZS5...
```

Secret 등록 결과

- 작성 방법은 ConfigMap과 유사
- stringData에 평문으로 입력하면 data 필드에 자동으로 Base64 로 변환

Secret 리소스 적용 방식 - 환경변수

```
apiVersion: v1
kind: Secret
metadata:
  name: mysql-secret
stringData:
  MYSQL_ROOT_PASSWORD: password123
...
```

```
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:
                secretKeyRef:
                  name: mysql-secret
                  key: MYSQL_ROOT_PASSWORD
```

Secret 리소스 적용 방식 - 볼륨 마운트

```
spec:
  containers:
  - name: k8s-backend-user
    image: solarhc/k8s-backend-user:0.2.4
    imagePullPolicy: Always
    ...
```

volumeMounts:

```
- name: backend-user-config-volume
  mountPath: /etc/config
  readOnly: true
```

```
- name: backend-user-secret-volume
  mountPath: /etc/secret
  readOnly: true
```

volumes:

```
- name: backend-user-config-volume
```

configMap:

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

```
- name: backend-user-secret-volume
```

secret:

```
  secretName: backend-user-secret
  items:
  - key: application-secret.yml
    path: application-secret.yml
```

deployment.yaml

Secret 실습

backend user : 볼륨 마운트 방식 Secret 작성 및 등록

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-user-secret
stringData:
  application-secret.yml: |
    spring:
      datasource:
        username: user
        password: 1234
```

입력을 통해 생성 파일을 통해 생성 서식화

현재 선택된 네임스페이스에 생성할 리소스를 명시하는

```
1  apiVersion: v1
2  kind: Secret
3  metadata:
4    name: backend-user-secret
5  stringData:
6    application-secret.yml: |
7      spring:
8        datasource:
9          username: user
10         password: 1234
11
```

업로드 Cancel

backend user : 기존 ConfigMap 에서 username, password 제거

Config Maps			
이름	레이블	생성 시간 ↑	
backend-user-config	-	11.hours.ago	⋮
		2.days.ago	Edit
			Delete

Edit a resource

YAML

JSON

```
18 f:application-dev.yml: {}
19 data:
20 application-dev.yml: [
21   server:
22     port: 8080
23
24   spring:
25     datasource:
26       url: jdbc:mysql://k8s-external-user-mysql-service:3306/user?serverTimezone=UTC&useSSL
27         =true&autoReconnect=true&useUnicode=true&characterEncoding=utf-8
28       driver-class-name: com.mysql.cj.jdbc.Driver
29     hikari:
30       connection-test-query: SELECT 1 # HikariCP 유효성 검사 추가
31       validation-timeout: 5000
32   jpa:
33     hibernate:
34       ddl-auto: create # 오직 테스트 환경에서만
35       generate-ddl: true # 오직 테스트 환경에서만 (spring.jpa.generate-ddl)
36       show-sql: true
37       open-in-view: false
38   kafka:
```

username: user
password: 1234 제거

spring:
datasource:
url: jdbc:mysql://k8s-external-user-mysql-service:3306/user?...
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

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

Update

Cancel

backend user : application-dev.yml 수정

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

application-dev.yml

backend user : Deployment 수정

```
spec:
  containers:
  - name: k8s-backend-user
    image: solarhc/k8s-backend-user:0.2.4
    imagePullPolicy: Always
    env:
    - name: SPRING_PROFILES_ACTIVE
      value: dev
    ports:
    - containerPort: 8080
  volumeMounts:
  - name: backend-user-config-volume
    mountPath: /etc/config
    readOnly: true
  - name: backend-user-secret-volume
    mountPath: /etc/secret
    readOnly: true
  volumes:
  - name: backend-user-config-volume
    configMap:
      name: backend-user-config
      items:
      - key: application-dev.yml
        path: application-dev.yml
  - name: backend-user-secret-volume
    secret:
      secretName: backend-user-secret
      items:
      - key: application-secret.yml
        path: application-secret.yml
```

Edit a resource

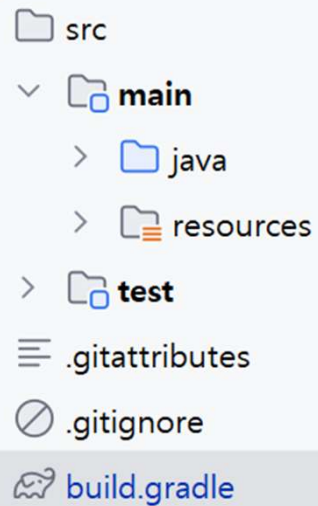
YAML	JSON
142 ▾	spec:
143 ▾	volumes:
144 ▾	- name: backend-user-config-volume
145 ▾	configMap:
146 ▾	name: backend-user-config
147 ▾	items:
148 ▾	- key: application-dev.yml
149 ▾	path: application-dev.yml
150 ▾	defaultMode: 420
151 ▾	- name: backend-user-secret-volume
152 ▾	secret:
153 ▾	secretName: backend-user-secret
154 ▾	items:
155 ▾	- key: application-secret.yml
156 ▾	path: application-secret.yml
157 ▾	defaultMode: 420
158 ▾	containers:
159 ▾	- name: k8s-backend-user
160 ▾	image: solarhc/k8s-backend-user:0.2.5
161 ▾	ports:
162 ▾	- containerPort: 8080

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

Update

Cancel

backend-user : 앱 버전 업그레이드



A file explorer view showing a 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 = '0.0.6'  
  
...
```

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
	solarhc/k8s-api-gateway:0.0.3	-	1 / 1	2 hours ago

97 spec:
98 replicas: 1
99 selector:
100 matchLabels:
101 app: k8s-backend-user
102 template:
103 metadata:
104 creationTimestamp: null
105 labels:
106 app: k8s-backend-user
107 spec:
108 containers:
109 - name: k8s-backend-user
110 image: solarhc/k8s-backend-user: 0.0.6
111 ports:

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

Update

Cancel

Scale

Edit

Restart

Delete