

아코디언 사용자 설명서

Version 2.10

목차

1. 소개
 - 1.1. 개요
 - 1.2. 시스템 구성
 - 1.3. 클러스터 구성
 - 1.4. 사용자와 권한
2. 로그인
 - 2.1. 웹 콘솔 로그인
 - 2.2. 계정
3. 화면 구성
 - 3.1. 화면 영역
 - 3.1.1. 헤더 영역
 - 3.1.1.1. 스코프 선택
 - 3.1.1.2. 사용자 설정
 - 3.1.2. 메뉴 영역
 - 3.1.2.1. 글로벌 메뉴
 - 3.1.2.2. 클러스터 메뉴
 - 3.1.2.3. 네임스페이스 메뉴
 - 3.1.3. 이벤트 영역
 - 3.1.3.1. 이벤트 영역의 최소화
 - 3.1.3.2. 이벤트 영역의 최대화
4. 메뉴
 - 4.1. 글로벌 메뉴
 - 4.1.1. 글로벌 대시보드
 - 4.1.2. 클러스터
 - 4.1.2.1. 대시보드
 - 4.1.3. 헬름
 - 4.1.3.1. 헬름 리포지터리
 - 4.1.4. 계정
 - 4.1.4.1. 사용자
 - 4.1.4.2. 그룹
 - 4.1.4.3. 권한
 - 4.1.4.4. 글로벌 멤버
 - 4.1.4.5. 사용자 접속 로그
 - 4.1.5. 글로벌 설정
 - 4.1.5.1. 활성화 키
 - 4.1.5.2. 글로벌 수신자
 - 4.2. 클러스터 메뉴
 - 4.2.1. 클러스터 대시보드
 - 4.2.2. 네임스페이스
 - 4.2.2.1. 네임스페이스 생성
 - 4.2.2.2. 네임스페이스 수정
 - 4.2.2.3. 네임스페이스 삭제
 - 4.2.3. 노드

- 4.2.3.1. Cordon
- 4.2.3.2. Drain
- 4.2.3.3. 라벨 수정
- 4.2.4. 애플리케이션
 - 4.2.4.1. 클러스터 카탈로그 템플릿
 - 4.2.4.2. 헬름
- 4.2.5. 빌드
 - 4.2.5.1. 클러스터 태스크 템플릿
 - 4.2.5.2. 클러스터 파이프라인 템플릿
- 4.2.6. 워크로드
 - 4.2.6.1. 워크로드 대시보드
 - 4.2.6.2. 파드
 - 4.2.6.3. 디플로이먼트
 - 4.2.6.4. 스테이트풀셋
 - 4.2.6.5. 데몬셋
 - 4.2.6.6. 레플리카셋
 - 4.2.6.7. 잡
 - 4.2.6.8. 크론잡
- 4.2.7. 구성
 - 4.2.7.1. 컨피그맵
 - 4.2.7.2. 시크릿
 - 4.2.7.3. HPA
 - 4.2.7.4. 리밋레인지
- 4.2.8. 네트워크
 - 4.2.8.1. 서비스
 - 4.2.8.2. 잉그레스
 - 4.2.8.3. 네트워크폴리시
- 4.2.9. 스토리지
 - 4.2.9.1. 퍼시스턴트볼륨
 - 4.2.9.2. 퍼시스턴트볼륨클레임
 - 4.2.9.3. 스토리지클래스
- 4.2.10. 커스텀 리소스
 - 4.2.10.1. 커스텀 리소스 생성
 - 4.2.10.2. 커스텀 리소스 수정
 - 4.2.10.3. 커스텀 리소스 삭제
- 4.2.11. 접근제어
 - 4.2.11.1. 룰
 - 4.2.11.2. 룰바인딩
 - 4.2.11.3. 클러스터롤
 - 4.2.11.4. 클러스터롤바인딩
 - 4.2.11.5. 서비스어카운트
- 4.2.12. 모니터링
 - 4.2.12.1. 시스템
 - 4.2.12.2. 이벤트 로그
 - 4.2.12.3. 컨테이너 로그
 - 4.2.12.4. 감사 로그
 - 4.2.12.5. 알림 로그
 - 4.2.12.6. 서비스메시

4.2.13. 설정

4.2.13.1. 클러스터 멤버

4.2.13.2. 클러스터 수신자

4.2.13.3. 알림 정책

4.2.13.4. 레지스트리

4.3. 네임스페이스 메뉴

4.3.1. 네임스페이스 대시보드

4.3.2. 애플리케이션

4.3.2.1. 카탈로그

4.3.3. 빌드

4.3.3.1. 파이프라인

4.3.3.2. 승인

4.3.3.3. 태스크 템플릿

4.3.4. 워크로드

4.3.5. 구성

4.3.6. 네트워크

4.3.7. 스토리지

4.3.8. 접근제어

4.3.9. 모니터링

4.3.9.1. 애플리케이션

4.3.10. 설정

4.3.10.1. 네임스페이스 수신자

4.3.10.2. 알림 정책

1. 소개

1.1. 개요

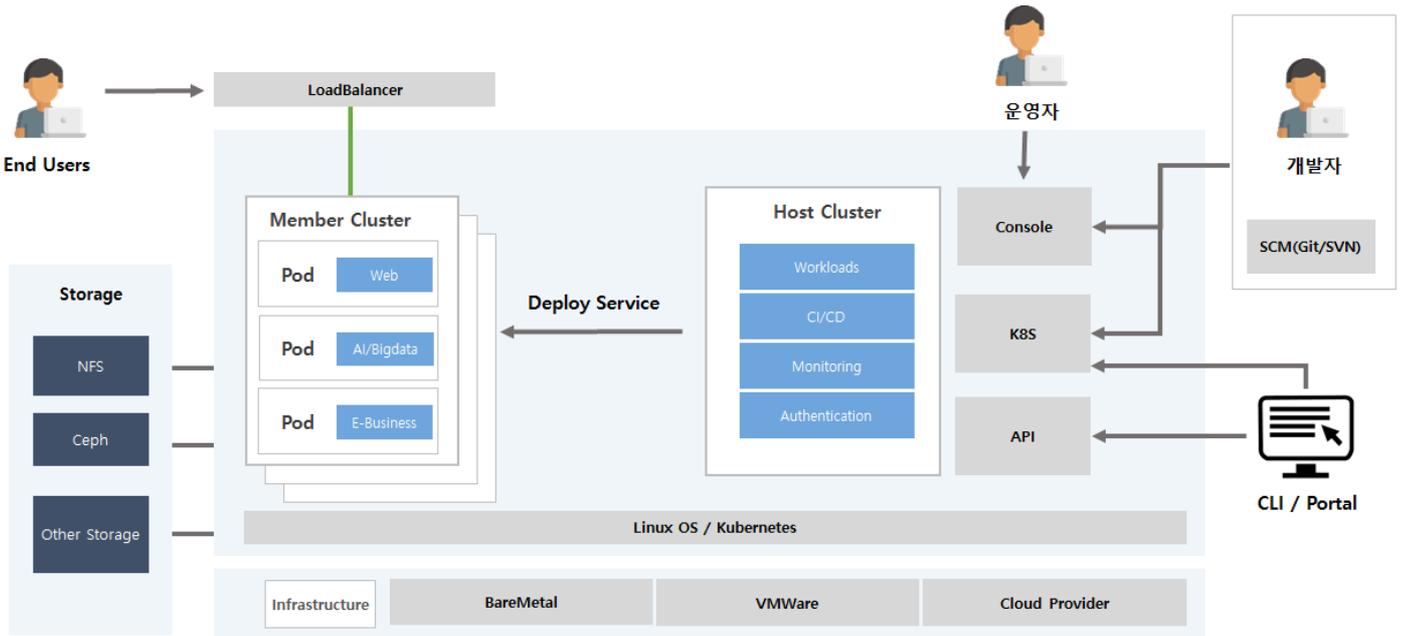
아코디언은 컨테이너 기반의 애플리케이션을 손쉽게 배포하고 자동화된 운영관리를 제공하기 위한 올인원 솔루션이다.

아코디언은 다음과 같은 주요기능을 가지고 있다.

- 멀티 클러스터 관리
- EKS/AKS/GCP/AWS/Azure 등 퍼블릭 클라우드 연동 및 관리
- CI/CD를 통한 애플리케이션 자동 배치 및 롤백
- 승인 파이프라인으로 빌드 자동화
- 글로벌 시스템 모니터링 및 WAS 모니터링
- 로그 및 감사 기능 제공
- Helm 제공 및 사용자 정의 컨테이너 관리

1.2. 시스템 구성

아코디언의 시스템 구성은 아래와 같다.



1.3. 클러스터 구성

아코디언은 멀티 클러스터 관리하며 클러스터는 역할별로 호스트/멤버 클러스터로 구분하여 관리한다.

구분	설명
호스트 클러스터	멀티 클러스터 관리 및 모니터링, 대시보드, 인증 기능을 제공한다.
멤버 클러스터	사용자의 서비스 애플리케이션이 배포되고 운영되는 클러스터이다.

1.4. 사용자와 권한

사용자는 스코프 별로 글로벌 권한, 클러스터 권한, 네임스페이스 권한 을 가질 수 있다.

“자세한 설명은 글로벌 권한/클러스터 권한/네임스페이스 권한을 참고한다.

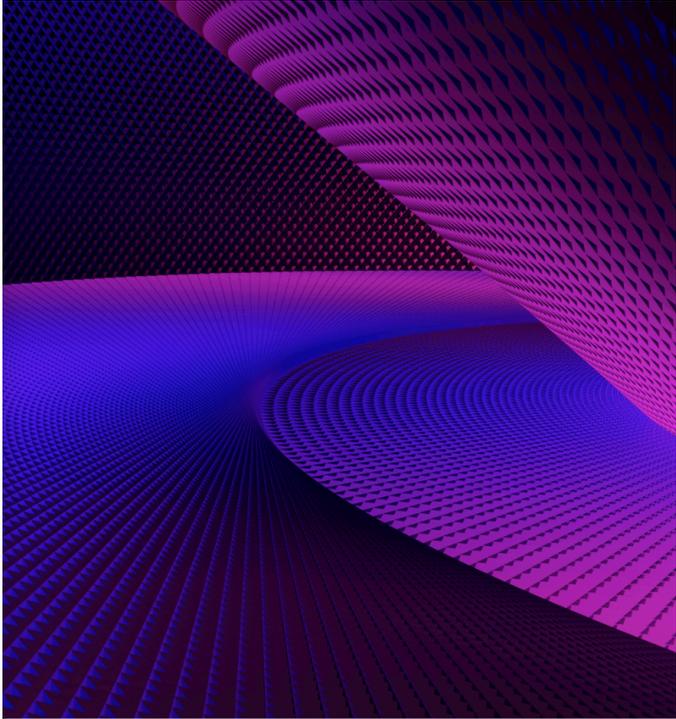
2. 로그인

2.1. 웹 콘솔 로그인

웹 브라우저에서 아코디언을 설치한 호스트 클러스터로 접속한다.

TIP

아코디언에서는 외부에서 접근할 수 있도록 30000 포트를 기본 제공한다.
이런 경우 웹 브라우저 접속시 `https://<HOST_IP>:30000` 을 사용한다.



All-in-One Solution

2.2. 계정

초기 계정 정보는 설치 매뉴얼을 참고한다.

3. 화면 구성

3.1. 화면 영역

화면은 다음과 같이 3개의 영역으로 나누어진다.

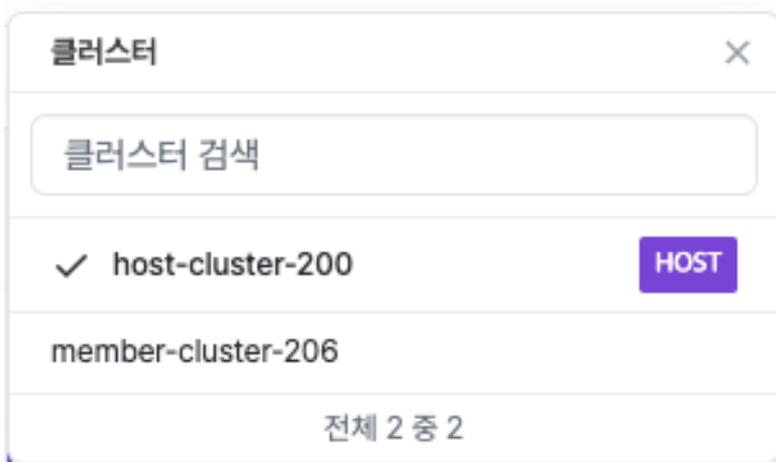
영역	설명
헤더 영역	화면의 상단에 위치한다. 스코프(클러스터/네임스페이스) 를 선택할 수 있고 사용자 정보가 표시된다.
메뉴 영역	화면의 좌측에 위치한다. 각 스코프(글로벌/클러스터/네임스페이스) 에 대한 메뉴를 표시한다. 메뉴는 사용자 권한에 따라 사용가능한 항목이 표시된다.
이벤트 영역	화면의 하단에 위치하여 멀티 클러스터에서 발생하는 이벤트를 표시한다.
메인 영역	각 메뉴에 대한 화면을 표시한다. 메뉴별 정보를 표시하고 기능을 설정, 제어, 모니터할 수 있다.

3.1.1. 헤더 영역



3.1.1.1. 스코프 선택

클러스터 선택



네임스페이스, 노드 관리와 같이 클러스터 스코프에서 필요한 작업을 수행하기 위해 클러스터를 선택할 수 있다.

네임스페이스 선택

네임스페이스 ✕

시스템 네임스페이스 OFF

네임스페이스 검색

✓ 전체 네임스페이스

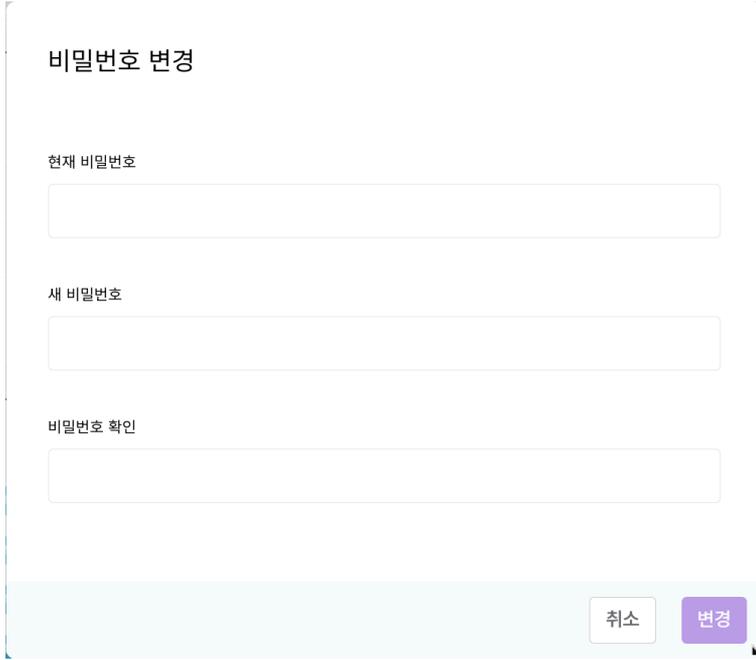
- backend
- cert-manager
- default
- frontend
- qa-test
- opensource

전체 7 중 7

애플리케이션, 빌드와 같이 네임스페이스 스코프에서 필요한 작업을 수행하기 위해 네임스페이스를 선택할 수 있다.

3.1.1.2. 사용자 설정

비밀번호 변경



비밀번호 변경

현재 비밀번호

새 비밀번호

비밀번호 확인

취소 변경

사용자의 비밀번호를 변경할 수 있다.

매뉴얼

아코디언 매뉴얼을 확인할 수 있다.

kubectl 사용 가이드



Kubectl 사용 가이드

Kubectl 다운로드

Windows Mac (Intel) Mac (Apple Silicon) Linux

```
curl.exe -LO "https://dl.k8s.io/release/v1.yy.z/bin/windows/amd64/kubectl.exe"
```

kubeconfig 다운로드

다운로드

- 아코디언에 버전에 맞는 kubectl 실행 파일 다운로드 경로를 알 수 있다.
- 아코디언 사용자의 kubeconfig 를 다운로드 받을 수 있다.

API Key

아코디언 API를 이용할 수 있는 token 을 관리한다.

API Key 목록

← 돌아가기

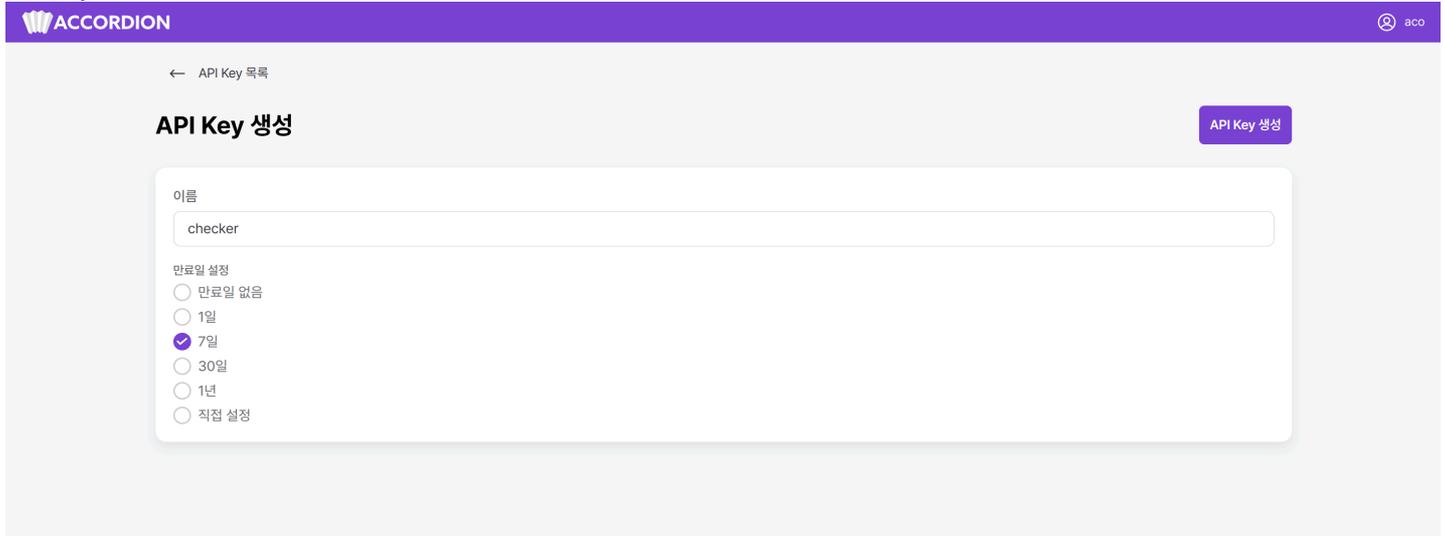
API Key API Key 생성

상태	이름	API Key	생성일 ↓	만료일
Active	sms	*****K2QA	2023-09-22, 09:52:14	2024-09-21, 09:52:14
Active	mail sender	acc-jtlq69692tulsQ423T091FmxudiuUln4jd5ucyayelo	2023-09-22, 09:49:44	2023-10-22, 09:49:44
Active	devops	*****VwoE	2023-09-22, 09:48:23	2023-09-29, 09:48:23
Active	monitoring	*****UvDA	2023-09-22, 09:47:47	2023-09-23, 09:47:47

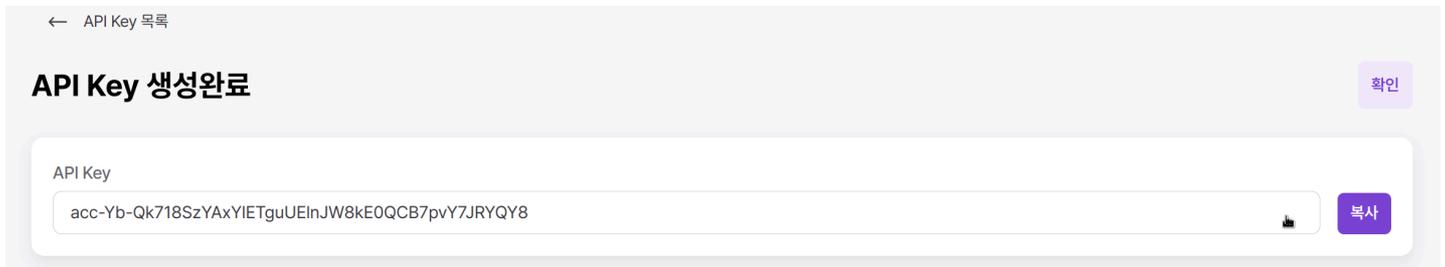
- 생성한 token 정보를 조회한다.

기능	설명
상태	token 이 유효한지 알 수 있다. (Active or InActive)
API Key	token 정보
생성일	token 생성일
만료일	token 만료일로 만료일이 되면 자동 삭제된다.

API Key 생성

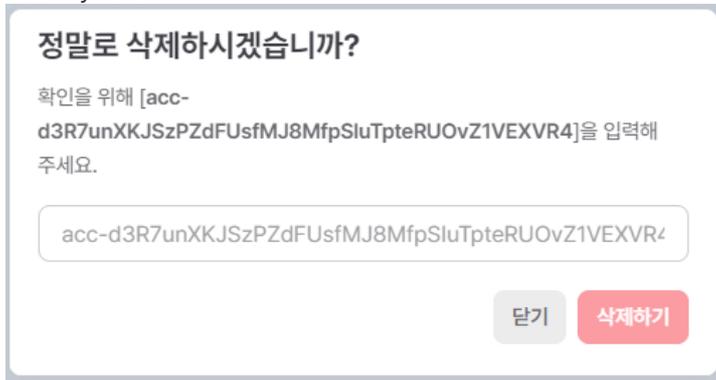


- 간단한 설명을 포함하여 token 을 생성한다.
- 가급적 만료일을 설정하여 생성하길 권장한다.



- 생성한 token 을 확인한다.

API Key 삭제



- 수동으로 token 을 삭제한다.

로그아웃

사용자 로그아웃을 한다.

3.1.2. 메뉴 영역

메뉴는 스코프별로 분류되며 스코프는 글로벌, 클러스터, 네임스페이스 3개로 구분된다.

3.1.2.1. 글로벌 메뉴

메뉴	설명
글로벌 대시보드	멀티 클러스터 모니터링 대시보드로 멀티 클러스터와 리소스의 상태 정보를 조회한다.
클러스터	호스트 클러스터 및 호스트 클러스터와 연결된 멤버 클러스터의 정보를 조회하고 관리한다.
헬름	헬름 차트를 사용하기 위해 헬름 리포지터리를 조회하고 관리한다.
계정	<p>아코디언 전반에 걸쳐 계정 정보를 관리한다.</p> <p>사용자, 그룹 및 그리고 글로벌/클러스터/네임스페이스 스코프 별 메뉴에 대한 권한을 관리하고 사용자/그룹과 글로벌 권한을 바인딩하여 글로벌 멤버로 설정한다.</p> <p>그 외에 사용자의 접속 로그를 조회할 수 있다.</p> <ul style="list-style-type: none"> • 사용자: 사용자 정보를 조회, 관리 • 그룹: 사용자 그룹 정보를 조회, 관리 • 글로벌 권한: 글로벌 스코프의 메뉴에 대해 사용 권한을 집합하여 단일 권한으로 관리 • 클러스터 권한: 클러스터 스코프의 메뉴에 대해 사용 권한을 집합하여 단일 권한으로 관리 • 네임스페이스 권한: 네임스페이스 스코프의 메뉴에 대해 사용 권한을 집합하여 단일 권한으로 관리 • 글로벌 멤버: 사용자/그룹과 글로벌 권한을 바인딩한다. • 사용자 접속 로그: 사용자의 접속 로그를 조회한다.
글로벌 설정	<p>활성화 키와 알림 발송을 위한 글로벌 수신자를 관리한다.</p> <ul style="list-style-type: none"> • 활성화 키: 활성화 키 정보를 조회, 등록한다. • 글로벌 수신자: 알림 발송을 받을 글로벌 수신자 정보를 관리한다.

3.1.2.2. 클러스터 메뉴

클러스터 메뉴는 개별 클러스터 스코프에서 서비스 배포와 운영을 위한 메뉴를 제공한다.

메뉴	설명
클러스터 대시보드	해당 클러스터의 리소스 상태 정보를 제공한다.
네임스페이스	해당 클러스터의 네임스페이스 정보를 관리한다.
노드	해당 클러스터를 구성하는 노드의 정보를 관리한다. 그리고 노드를 스케줄 대상에서 제외하거나 노드에 배포된 파드를 다른 노드로 이동시킬 수 있다.
애플리케이션	헬름을 이용해 애플리케이션을 클러스터에 배포한다.
워크로드	워크로드 리소스 정보를 관리하기 위한 대시보드를 제공하고 편집기 등으로 관리한다. <ul style="list-style-type: none"> 워크로드 대시보드: 배포한 파드 정보를 차트로 표시한다. 파드: 파드 정보를 제공하고 관리한다. 컨테이너 로그 및 터미널을 이용해 접속할 수 있다. 디플로이먼트/스테이트풀셋: 파드를 관리하는 리소스로 편집기로 관리하며 오토스케일을 설정할 수 있다. 데몬셋: 노드마다 파드를 실행하도록 제어한다. 잡/크론잡: 특정 작업을 일시적/주기적으로 실행할 수 있도록 관리한다.
구성	시스템과 애플리케이션 구성 및 설정에 필요한 정보를 관리한다. <ul style="list-style-type: none"> 컨피그맵/시크릿: 일반 정보 또는 기밀성 정보를 저장한다. HPA/리미트레이저: 수평적 오토스케일 및 리소스 할당을 제한한다.
네트워크	애플리케이션 연동을 위해 네트워크 구성에 필요한 리소스를 관리한다. <ul style="list-style-type: none"> 서비스: 애플리케이션을 네트워크 서비스로 노출한다. 인그레스: 서비스를 http 또는 https 경로로 노출한다. 네트워크폴리시: OSI 3 또는 4 계층에서 트래픽 흐름을 제어한다.
스토리지	스토리지 목록을 확인할 수 있으며 YAML/생성/수정/삭제할 수 있다. <ul style="list-style-type: none"> 퍼시스턴트볼륨: NFS 와 같이 스토리지 구현에 필요한 정보를 관리한다. 퍼시스턴트볼륨클레임: 스토리지에 대해 요청을 하는 리소스 정보를 관리한다. 스토리지클래스: 퍼시스턴트볼륨을 동적으로 배포한다.
커스텀 리소스	쿠버네티스 커스텀 리소스를 관리한다.
접근제어	쿠버네티스의 역할 기반 접근 제어를 관리하고 구성한다. <ul style="list-style-type: none"> 서비스어카운트: 주로 파드에 인증 정보를 제공한다. (사용자 역할) 롤/클러스터롤: API나 리소스에 대한 권한을 설정한다. 롤바인딩/클러스터롤바인딩: 롤/클러스터롤을 사용자 또는 서비스어카운트 등에 연결한다.

<p>모니터링</p>	<p>시스템 상태 및 각종 로그를 조회한다.</p> <ul style="list-style-type: none"> • 시스템: 클러스터 내 CPU, 메모리와 같은 리소스 사용량 정보를 제공한다. • 이벤트 로그: 쿠버네티스 이벤트 로그를 조회한다. • 컨테이너 로그: 쿠버네티스에 배포되었던 컨테이너에서 발생한 로그를 조회한다. • 감사 로그: 클러스터 내 감사 로그를 조회한다. • 알림 로그: 클러스터 내 발생한 알림 정보를 조회한다. • 서비스메시: 클러스터 내 서비스 간의 통신 정보를 표시한다.
<p>설정</p>	<p>클러스터를 사용하는 멤버를 구성하고 알림 정책을 세운다. 레지스트리 정보 및 알림을 위한 클러스터 수신자 정보를 설정한다.</p> <ul style="list-style-type: none"> • 클러스터 멤버: 클러스터 권한과 사용자/그룹을 연결한다. • 클러스터 수신자: 알림 발송을 받을 클러스터 수신자 정보를 관리한다. • 알림 정책: 클러스터 알림 발송 규칙 및 발송 주기/수신자 등을 설정한다. • 레지스트리: 컨테이너 이미지를 배포하고 가져올 레지스트리 저장소 정보를 설정한다.

3.1.2.3. 네임스페이스 메뉴

네임스페이스 메뉴는 네임스페이스 스코프에서 서비스의 배포와 운영을 위한 메뉴를 제공한다.

메뉴 중 일부는 클러스터 메뉴와 중복된다. 중복되는 메뉴의 경우 동일한 기능에 대해 접근하는 스코프의 차이를 제외하고 대부분 동일한 기능을 제공한다.

메뉴	설명
네임스페이스 대시보드	해당 네임스페이스의 리소스 상태 정보를 제공한다.
애플리케이션	<p>헬름과 카탈로그를 이용해 애플리케이션을 클러스터에 배포한다.</p> <ul style="list-style-type: none"> • 헬름: 헬름 차트로 애플리케이션을 배포한다. 주로 빌드를 수행하지 않는 서비스형 애플리케이션을 배포한다. • 카탈로그: 카탈로그로 애플리케이션을 배포한다. 빌드 파이프라인을 지원한다.
빌드	<p>애플리케이션 빌드에 필요한 정보를 제어하고 관리한다. 컨테이너 이미지를 빌드할 수 있고 빌드에 필요한 파이프라인 또는 승인에 대해 설정한다.</p> <ul style="list-style-type: none"> • 파이프라인: 태스크를 조합하여 빌드를 수행할 파이프라인을 작성한다. • 승인: 파이프라인에서 발생한 승인 요청을 처리한다. • 태스크 템플릿: 파이프라인 작성시 사용하는 태스크에 대한 템플릿을 관리한다.
워크로드	클러스터 메뉴 의 워크로드 메뉴와 동일한 기능을 네임스페이스 스코프에서 수행한다.
구성	클러스터 메뉴 의 구성 메뉴와 동일한 기능을 네임스페이스 스코프에서 수행한다.
네트워크	클러스터 메뉴 의 네트워크 메뉴와 동일한 기능을 네임스페이스 스코프에서 수행한다.
스토리지	퍼시스턴트볼륨클레임 리소스 정보를 관리한다.
접근제어	<p>쿠버네티스의 역할 기반 접근 제어를 관리하고 구성한다.</p> <ul style="list-style-type: none"> • 서비스어카운트: 주로 파드에 인증 정보를 제공한다. (사용자 역할) • 룰: API나 리소스에 대한 권한을 설정한다. • 룰바인딩: 룰을 사용자 또는 서비스어카운트 등에 연결한다.
모니터링	<p>시스템 및 애플리케이션 상태 및 각종 로그를 조회한다.</p> <ul style="list-style-type: none"> • 시스템: 네임스페이스 내 CPU, 메모리와 같은 리소스 사용량 정보를 제공한다. • 애플리케이션: 스카우터로 수집한 애플리케이션에 대한 메트릭 정보를 제공한다. • 이벤트 로그: 네임스페이스에서 발생한 쿠버네티스 이벤트 로그를 조회한다. • 컨테이너 로그: 네임스페이스에 배포되었던 컨테이너에서 발생한 로그를 조회한다. • 감사 로그: 네임스페이스 내 감사 로그를 조회한다. • 알림 로그: 네임스페이스 내 발생한 알림 정보를 조회한다.

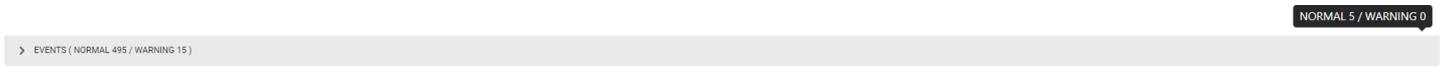
설정	<p>네임스페이스를 사용하는 멤버를 구성한다. 그리고 알림 정책을 세우고 알림을 위한 네임스페이스 수신자 정보를 설정한다.</p> <ul style="list-style-type: none">• 네임스페이스 멤버: 네임스페이스 권한과 사용자/그룹을 연결한다.• 네임스페이스 수신자: 알림 발송을 받을 네임스페이스 수신자 정보를 관리한다.• 알림 정책: 네임스페이스 알림 발송 규칙 및 발송 주기/수신자 등을 설정한다.
----	---

3.1.3. 이벤트 영역

이벤트는 멀티 클러스터 쿠버네티스 실시간 이벤트 정보를 제공하며 기본으로 최소화되어있으며 이벤트 영역을 클릭하면 상세 조회할 수 있다.

3.1.3.1. 이벤트 영역의 최소화

이벤트 영역에 우측 톨팁으로 현재 실시간으로 발생한 이벤트 수를 확인할 수 있다.



3.1.3.2. 이벤트 영역의 최대화

이벤트 영역의 최대화 시 이벤트 상세 확인 및 검색할 수 있다.

The screenshot shows a table of events with columns: Age, 클러스터 (host-cluster), 네임스페이스 (mj/test), 타입 (Normal), 원인 (Pulling), 오브젝트 (pod/test-27500090-1-t2thk), and 메세지 (Pulling image "busybox").

항목	설명
Age	이벤트가 발생한 시간
클러스터	클러스터 명
네임스페이스	네임스페이스 명
타입	이벤트 타입 <ul style="list-style-type: none"> Normal: 일반적인 작업으로 발생한 이벤트 Warning: 오류에 의해 발생한 이벤트
원인	이벤트 발생 이유
오브젝트	오브젝트 정보
메세지	이벤트 상세 메시지

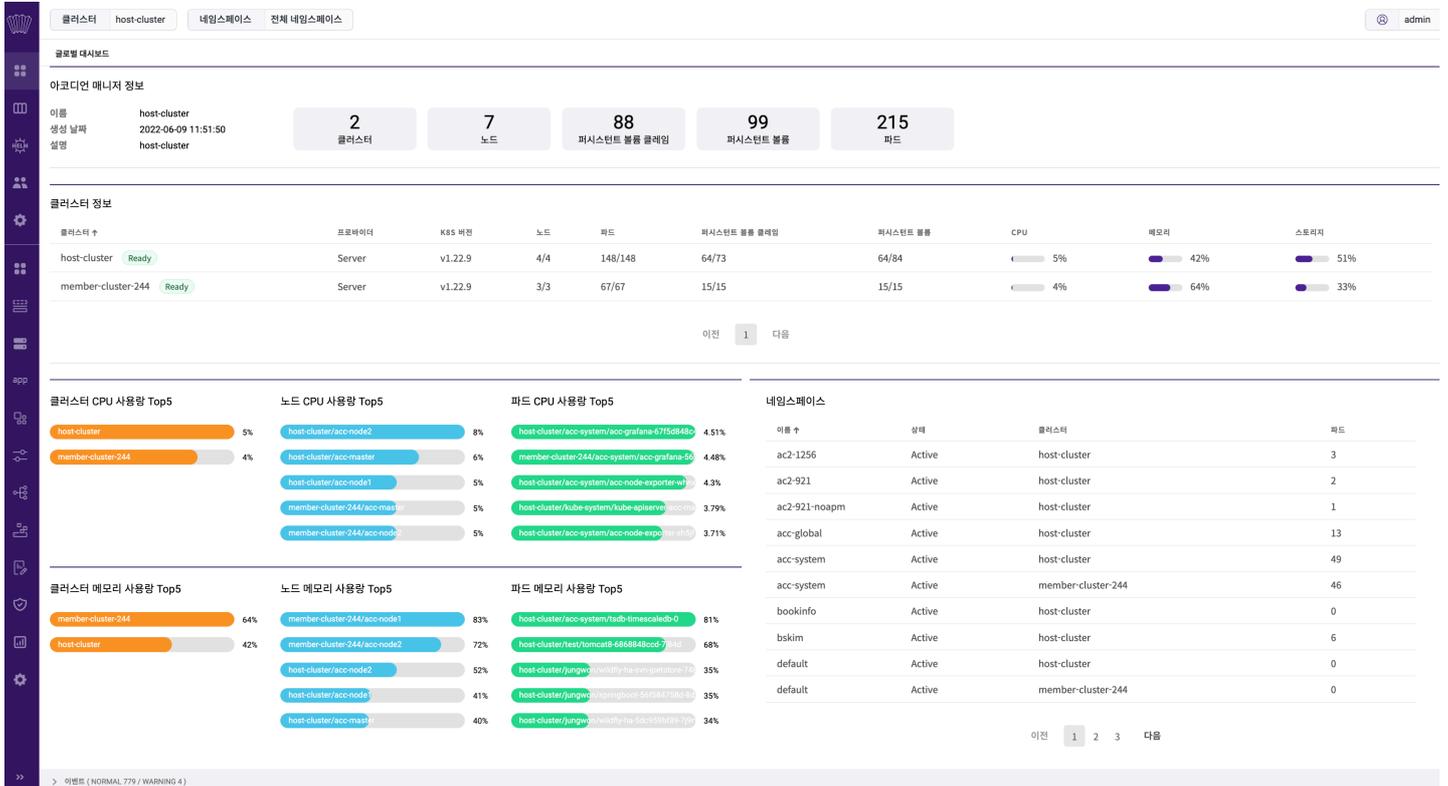
4. 메뉴

글로벌/클러스터/네임스페이스 각 스코프에 대한 메뉴를 설명한다.

4.1. 글로벌 메뉴

4.1.1. 글로벌 대시보드

글로벌 대시보드는 쿠버네티스 리소스 및 CPU/메모리와 같은 시스템 리소스 사용 현황 정보를 제공한다.



제공하는 정보는 다음과 같다.

항목	설명
아코디언 매니저 정보	호스트 클러스터 설명 및 리소스 정보를 제공한다.
클러스터 정보	아코디언과 연결된 클러스터들의 시스템/리소스 정보를 제공한다.
사용량 Top5	클러스터, 노드, 파드 별로 CPU/메모리 사용량 상위 5개에 대한 정보를 차트로 제공한다.
네임스페이스	모든 클러스터에 생성된 네임스페이스 정보를 제공한다.

4.1.2. 클러스터

클러스터는 컨테이너화된 애플리케이션을 실행하기 위한 노드의 집합을 의미한다. 컨테이너화된 애플리케이션은 개별 노드에 얽매이지 않고 클러스터에 배포된다. 클러스터는 클러스터를 관리하는 호스트 클러스터와 사용자의 애플리케이션이 배포되는 멤버 클러스터로 구분되며 호스트 클러스터는 멤버 클러스터의 역할도 수행한다. 아코디언은 하나의 호스트 클러스터와 하나 이상의 멤버 클러스터로 구성된다.

TIP

호스트/멤버 클러스터의 구성 예

- 싱글 클러스터 구성
 - 하나의 클러스터가 호스트/멤버 클러스터 역할을 모두 수행
- 관리 클러스터와 운영 클러스터 분리
 - 호스트 클러스터에서는 관리의 업무만 수행하고 별도의 멤버 클러스터에만 애플리케이션을 배포

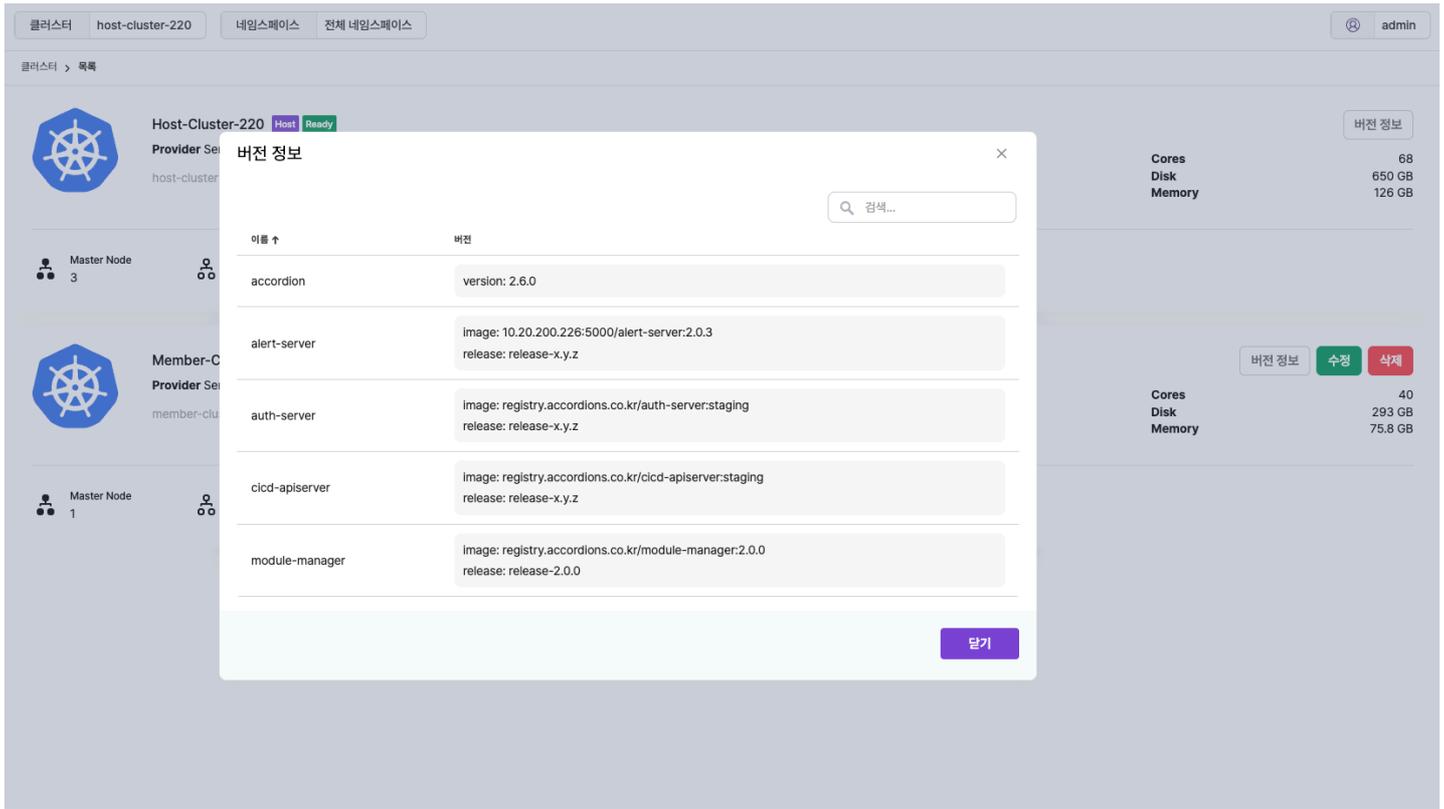
4.1.2.1. 대시보드

클러스터 메뉴에서는 호스트 클러스터의 정보를 조회하고 멤버 클러스터를 관리할 수 있다.

항목	설명
Cluster Name	클러스터 이름
Provider	서비스 제공자 (e.g. EKS, AKS, GKE..)
K8sVersion	쿠버네티스 버전
Created	클러스터 생성 시간 정보
Cores, Disk, Memory	클러스터 전체 리소스(CPU/디스크/메모리) 정보
Master Node, Worker Node	클러스터를 구성하는 마스터 노드와 워커 노드의 개수
Namespaces, Pods	클러스터에 생성된 네임스페이스와 파드의 개수
버전 정보	모듈별 버전 정보 제공
수정, 삭제	멤버 클러스터에만 있는 기능으로 클러스터 정보를 수정 또는 삭제

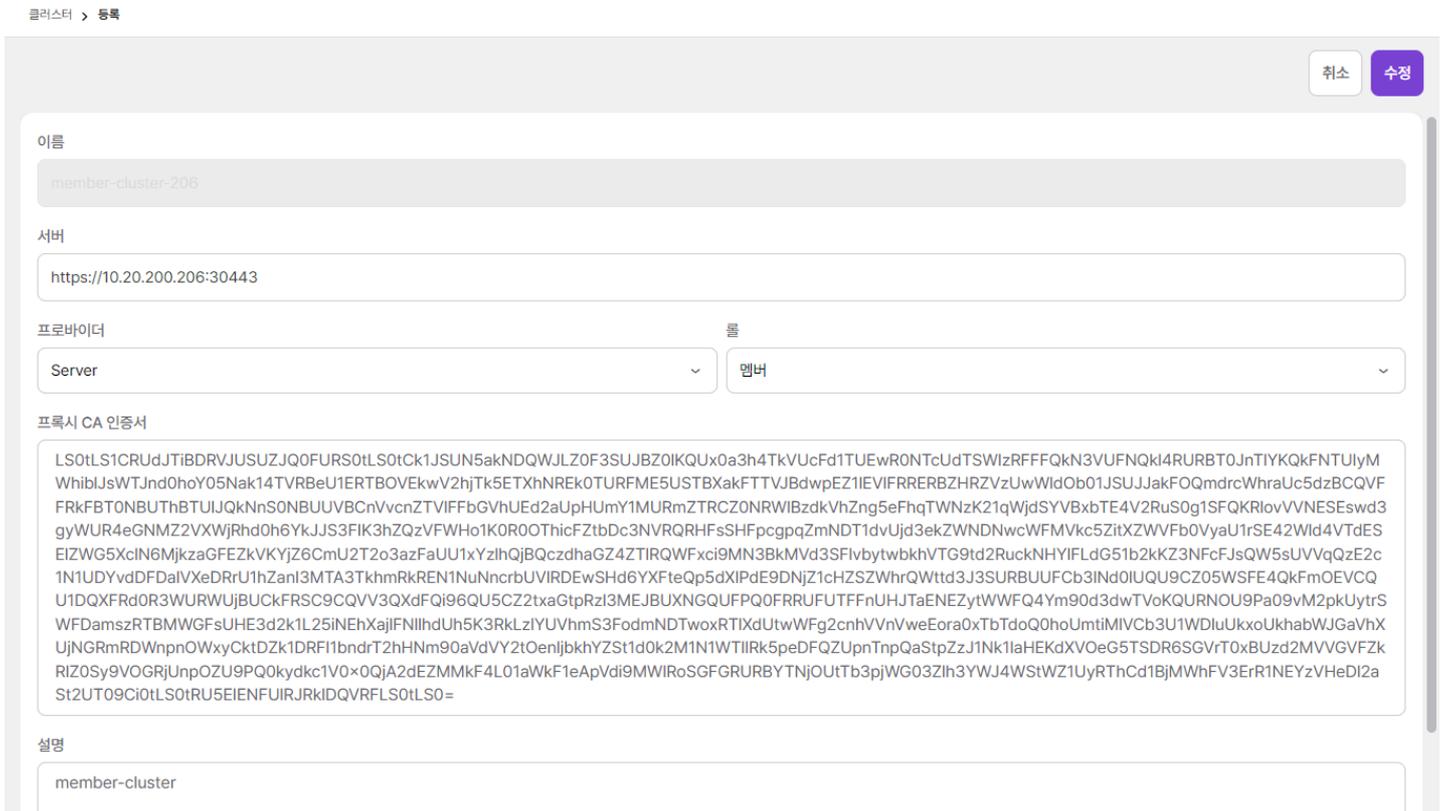
버전 정보

클러스터에 등록된 모듈들의 버전 정보를 보여준다.



멤버 클러스터 정보 수정

클러스터 목록에 있는 멤버 클러스터는 정보를 수정할 수 있다. 클러스터 정보를 수정하려면 클러스터 카드 우측의 수정 버튼을 선택하면 클러스터 수정 화면으로 이동한다.



제공하는 정보는 다음과 같다.

항목	설명
이름	클러스터 이름 (변경 불가)
서버	멤버 클러스터에 설치된 member-agent 의 엔드포인트를 입력
프로바이더	서비스 제공자 (e.g. EKS, AKS, GKE..)
롤	클러스터의 역할로 Member 만 선택 가능
프록시 CA 인증서	member-agent 에서 사용하는 CA 인증서
설명	클러스터 설명

멤버 클러스터 삭제

멤버 클러스터를 관리 목록에서 제거하려면 클러스터 카드 우측의 삭제 버튼을 선택한다. 클러스터 삭제는 아코디언의 관리 목록에서 제외하는 것이지만 실제 클러스터를 삭제하지는 않는다.

클러스터 | host-cluster-220 | 네임스페이스 | 전체 네임스페이스

admin

클러스터 > 목록



Host-Cluster-220 Host Ready

Provider Server **K8sVersion V1.26.6** Created 2023. 6. 8. 오후 4:57:28

host-cluster

버전 정보

Cores	68
Disk	650 GB
Memory	126 GB

Master Node 3

Worker Node 3

Namespaces 42

Pods 201



Member-Cluster-226 Member Ready

Provider Server **K8sVersion V1.26.6** Created 2023. 6. 8. 오후 4:57:43

member-cluster

버전 정보

수정	삭제
Cores	40
Disk	293 GB
Memory	75.8 GB

Master Node 1

Worker Node 2

Namespaces 12

Pods 110

4.1.3. 헬름

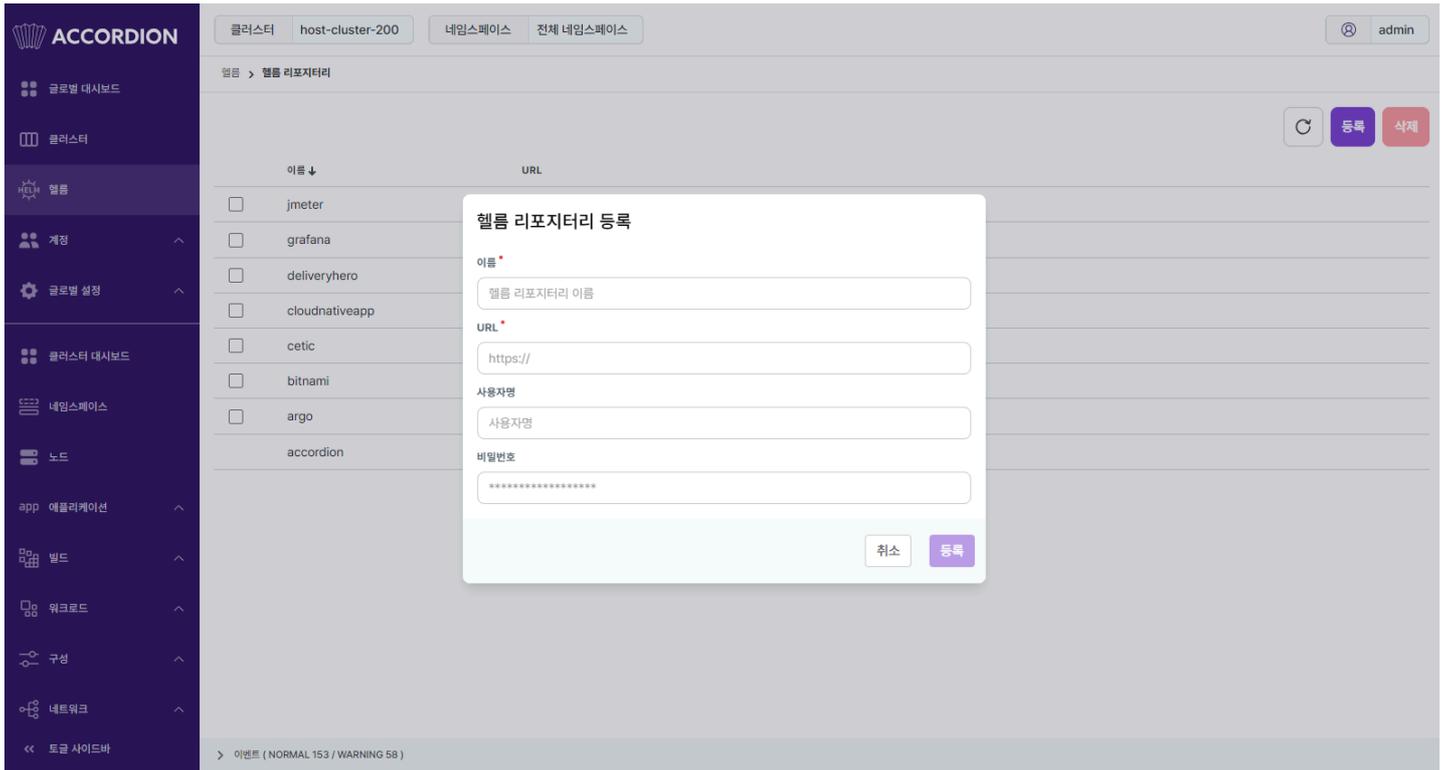
헬름은 쿠버네티스 애플리케이션 배포를 지원하는 도구로 아코디언에서는 헬름 리포지터리 및 차트를 관리하고 이를 이용해 클러스터에 애플리케이션을 배포할 수 있게 서비스를 제공한다.

아코디언은 폐쇄망에서도 헬름 차트를 관리할 수 있도록 accordion 이라는 기본 리포지터리를 제공한다. 사용자가 외부의 리포지터리를 추가하기 위해서는 인터넷을 통해 외부에 접근이 가능해야 한다.

4.1.3.1. 헬름 리포지터리

헬름 리포지터리 등록

헬름 메뉴에 접근하여 상단의 등록 버튼을 선택하면 헬름 리포지터리 정보를 입력하는 모달 화면이 표시된다.



항목	설명
이름	헬름 리포지터리 이름 (필수)
URL	헬름 리포지터리 URL (필수)
사용자명	헬름 리포지터리 아이디 (옵션)
비밀번호	헬름 리포지터리 패스워드 (옵션)

헬름 리포지터리 업데이트

우측 상단의 업데이트 버튼을 선택하면 헬름 리포지터리의 차트 정보를 최신으로 업데이트할 수 있다.

헬름 > 헬름 리포지터리





이름 ↓	URL
<input type="checkbox"/> jmeter	https://liukunup.github.io/helm-charts
<input type="checkbox"/> grafana	https://grafana.github.io/helm-charts
<input type="checkbox"/> deliveryhero	https://charts.deliveryhero.io/
<input type="checkbox"/> cloudfnativeapp	https://cloudnativeapp.github.io/charts/curated/
<input type="checkbox"/> cetic	https://cetic.github.io/helm-charts
<input type="checkbox"/> bitnami	https://charts.bitnami.com/bitnami
<input type="checkbox"/> argo	https://argoproj.github.io/argo-helm
accordion	http://chartmuseum-chartmuseum.acc-global.svc.cluster.local:8080/charts

이전 1 다음

헬름 리포지터리 삭제

삭제하려는 리포지터리를 선택하고 우측 상단의 삭제 버튼을 선택한다.

NOTE

기본 제공하는 accordion 리포지터리는 삭제할 수 없다.

헬름 > 헬름 리포지터리





이름 ↓	URL
<input checked="" type="checkbox"/> jmeter	https://liukunup.github.io/helm-charts
<input type="checkbox"/> grafana	https://grafana.github.io/helm-charts
<input type="checkbox"/> deliveryhero	https://charts.deliveryhero.io/
<input type="checkbox"/> cloudfnativeapp	https://cloudnativeapp.github.io/charts/curated/
<input type="checkbox"/> cetic	https://cetic.github.io/helm-charts
<input type="checkbox"/> bitnami	https://charts.bitnami.com/bitnami
<input type="checkbox"/> argo	https://argoproj.github.io/argo-helm
accordion	http://chartmuseum-chartmuseum.acc-global.svc.cluster.local:8080/charts

이전 1 다음

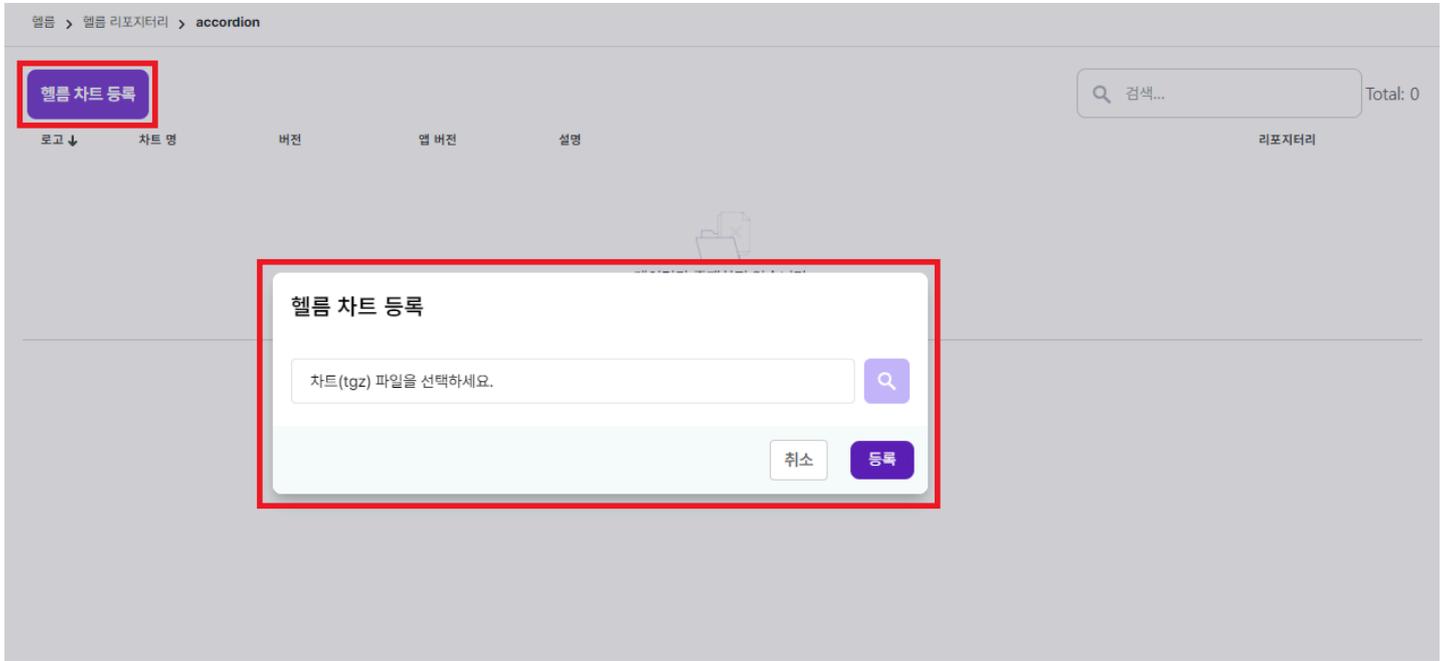
헬름 차트

헬름 차트 등록

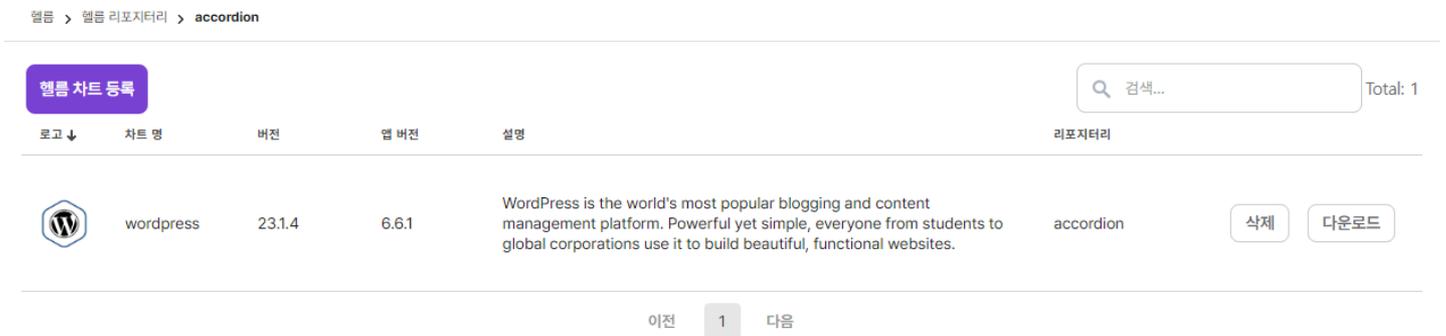
아코디언에서는 기본 제공하는 accordion 리포지터리에만 헬름 차트를 등록할 수 있다. 헬름 차트를 등록하기 위해 accordion 리포지터리를 선택하고 상단의 헬름 차트 등록 버튼을 선택한다. 모달에서 헬름 차트 파일을 입력하여 등록한다.

NOTE

아코디언에 헬름 차트를 등록하기 전에 tgz 확장자로 압축한 헬름 차트 파일을 미리 준비한다.

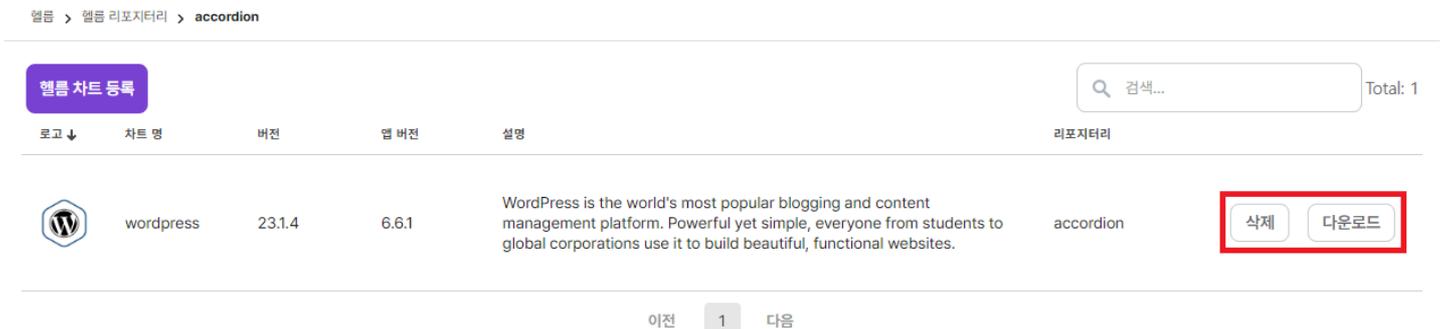


헬름 차트가 정상적으로 등록되면 아래와 같이 조회할 수 있다.



헬름 차트 삭제와 다운로드

헬름 차트 목록 우측의 버튼을 이용해 차트를 삭제하거나 파일로 다운로드 받을 수 있다.

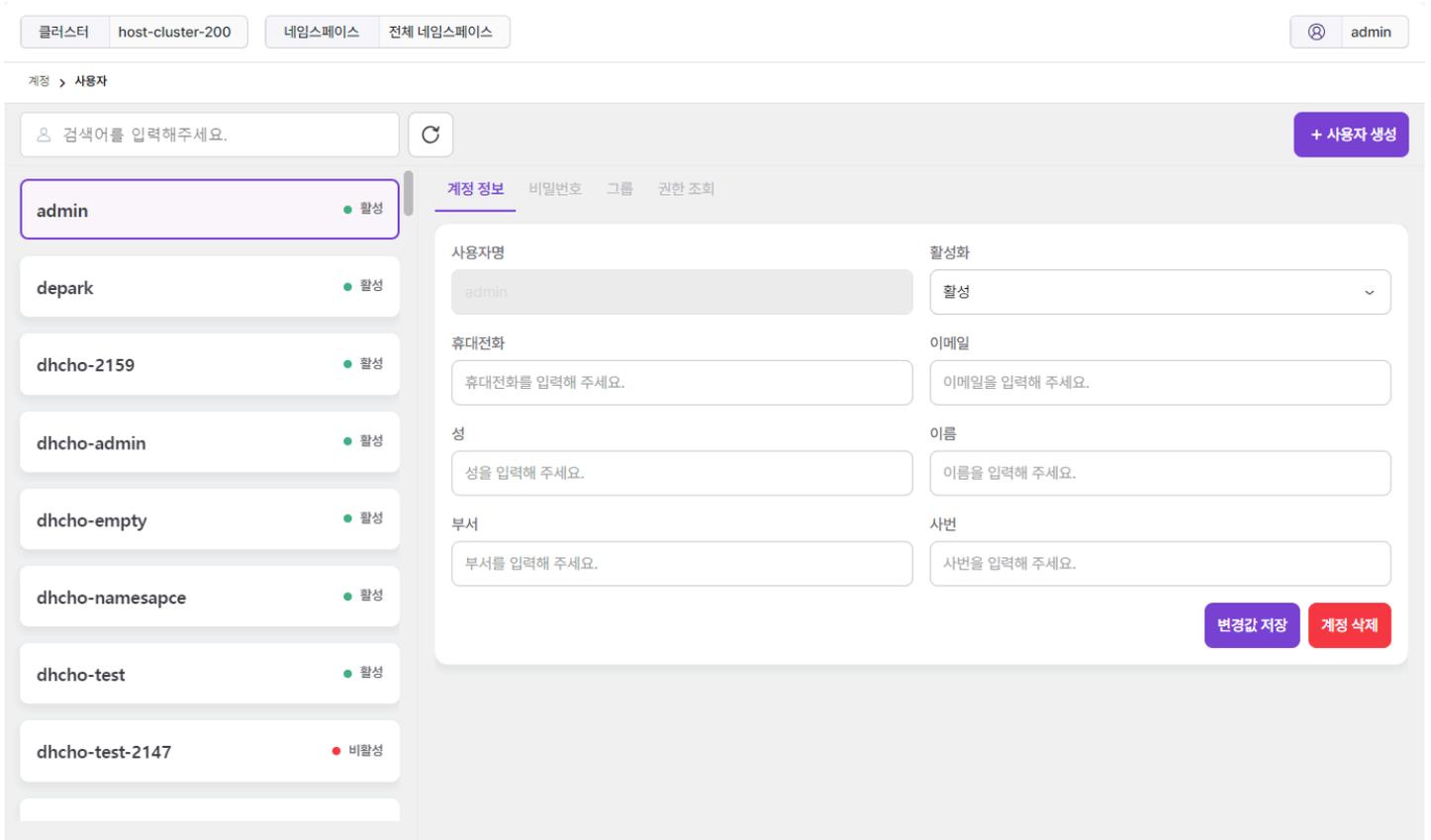


4.1.4. 계정

계정에서는 사용자 정보와 권한, 접속 로그와 같이 전반적인 계정 관리에 필요한 기능을 제공한다.

4.1.4.1. 사용자

사용자 메뉴에서 사용자 정보를 등록하거나 수정/삭제할 수 있다. 사용자 목록 정보에서는 사용자의 활성화 여부에 따라 목록의 아이콘이 녹색 또는 빨간색으로 표시된다.



사용자 페이지는 계정 정보, 비밀번호, 그룹, 권한 조회 으로 이뤄진 4개의 탭으로 구성되어 있으며, 제공하는 정보는 다음과 같다.

계정 정보

항목	설명
사용자명	사용자 아이디 (영문, 숫자, 하이픈('-'), 언더스코어('_'), 마침표('.')만 입력 가능하며, 시작과 끝에 영문 또는 숫자만 허용)
활성화	사용자 활성화 여부 (활성 또는 비활성)
휴대전화	사용자 휴대전화 번호(숫자만 입력 가능하며 10~11자리 입력)
이메일	사용자 이메일 주소
성	사용자 성
이름	사용자 이름
부서	사용자 부서명
사번	사용자 사원 번호 (영문, 숫자, 하이픈('-')만 입력 가능)

비밀번호

항목	설명
새 비밀번호	사용자 비밀번호를 입력 (영문, 숫자, 특수문자로 입력하며 8~16 자리로 특수문자가 1자 이상 포함)
새 비밀번호 확인	새 비밀번호와 동일하게 사용자 비밀번호를 다시 입력
비밀번호 초기화	비밀번호 초기화 여부 (활성 또는 비활성)

그룹

항목	설명
그룹	사용자와 그룹을 다대일의 관계로 연결

권한 조회

계정 > 사용자

C
+ 사용자 생성

admin
● 활성

viewer
● 활성

guest
● 활성

계정 정보
비밀번호
그룹
권한 조회

개인
내보내기

글로벌 권한

global-admin	global-viewer
✓	

클러스터 권한

클러스터 이름	cluster-admin	cluster-viewer
stg-host-cluster		
stg-member-cluster		

네임스페이스 권한

클러스터 이름	네임스페이스 이름	namespace-admin	namespace-viewer
	acc-global		
	acc-system		
	default		
	monitoring		

항목	설명
권한 조회 유형	전체 (개인+그룹), 개인, 그룹 으로 구성되었으며, 특정 유형을 선택하여 조회할 수 있다.
내보내기	권한을 csv 파일로 내보낸다.
글로벌 권한 클러스터 권한 네임스페이스 권한	사용자가 해당 권한을 가지고 있다면 v 로 체크된다. 상단의 권한 이름을 선택할 경우 해당 권한 페이지로 이동된다.

https://10.60.200.221:30000/docs/

27/349

사용자 등록

사용자를 등록하기 위해서는 우측 상단에 있는 + 사용자 생성 버튼을 선택하면, 사용자 등록 페이지로 이동한다.

붉은점으로 표시된 정보는 필수 사항이며, 정보를 입력한 후 사용자 생성 버튼을 선택한다.

← 사용자 목록 사용자 생성

사용자명 *
사용자명을 입력해 주세요.

비밀번호 *
비밀번호를 입력해 주세요.

비밀번호 확인 *
다시 한번 비밀번호를 입력해 주세요.

활성화 *
비활성

비밀번호 초기화 * ①
비활성

이메일
이메일을 입력해 주세요.

휴대전화
휴대전화를 입력해 주세요.

성
성을 입력해 주세요.

이름
이름을 입력해 주세요.

부서
부서를 입력해 주세요.

사번
사번을 입력해 주세요.

그룹
그룹 추가

사용자 수정

사용자를 수정하기 위해서는 사용자 목록에서 사용자를 선택하고, 이후 수정할 정보를 입력한 후 변경값 저장 버튼을 선택한다.

사용자 정보 수정은 각 탭 별로 수정된다.

사용자 삭제

사용자를 삭제하기 위해서는 목록에서 삭제할 대상이 되는 사용자를 선택하고 계정 정보 탭의 계정 삭제 버튼을 선택한다.

이후 나타나는 모달에서 사용자명을 입력한 후 삭제하기 버튼을 선택하면 완료된다.

계정 > 사용자

검색어를 입력해주세요. + 사용자 생성

admin ● 활성

depark ● 활성

dhcho-2159 ● 활성

dhcho-admin ● 활성

dhcho-empty ● 활성

dhcho-namesapce ● 활성

dhcho-test ● 활성

dhcho-test-2147 ● 비활성

계정 정보 | 비밀번호 | 그룹 | 권한 조회

사용자명: admin

활성화: 활성

휴대전화:

이메일: 이메일을 입력해 주세요.

이름: 이름을 입력해 주세요.

사번: 사번을 입력해 주세요.

정말로 삭제하시겠습니까?
확인을 위해 admin을 입력해 주세요.

admin

닫기 **삭제하기**

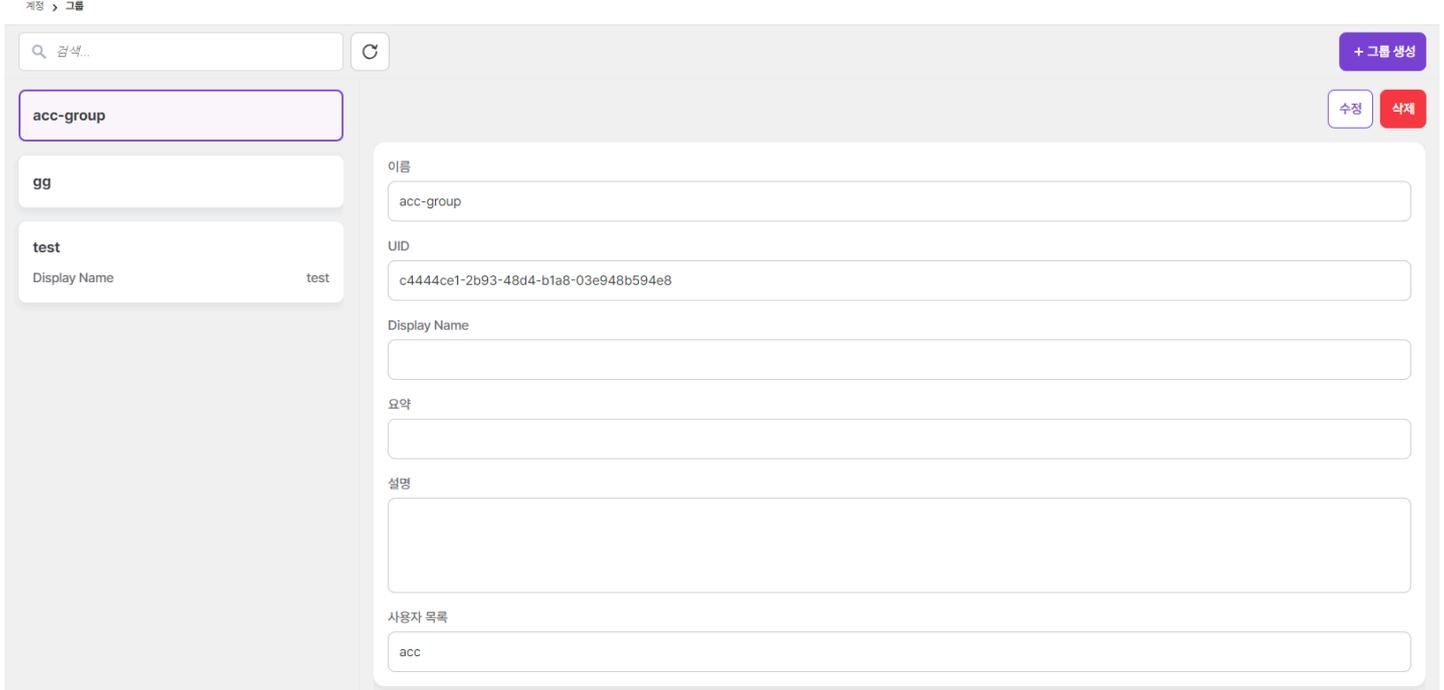
변경값 저장 **계정 삭제**

4.1.4.2. 그룹

그룹 메뉴에서 그룹 정보를 등록, 수정, 삭제할 수 있다. 관리자는 멤버 메뉴에서 사용자 또는 그룹에 권한을 부여할 수 있다. 그룹 정보에서는 해당 그룹에 등록된 사용자 목록 정보를 제공한다.

NOTE

그룹 은 사용자 집합을 의미한다.



항목	설명
이름	그룹 이름
UID	그룹의 고유 아이디
Display Name	사용자, 글로벌 멤버, 클러스터 멤버, 네임스페이스 멤버에서 해당 그룹이 표시되는 이름
요약	그룹의 요약된 정보
설명	그룹의 상세 설명
사용자 목록	그룹과 연결된 사용자의 목록 (사용자 이름)

그룹 등록

그룹을 등록하기 위해서는 우측 상단의 + 그룹 생성 버튼을 선택한 후 필수 항목인 이름 및 기타 선택항목을 작성 후 그룹 생성 버튼을 선택한다.

← 그룹 목록

그룹 생성

이름 *

Display Name

요약

설명

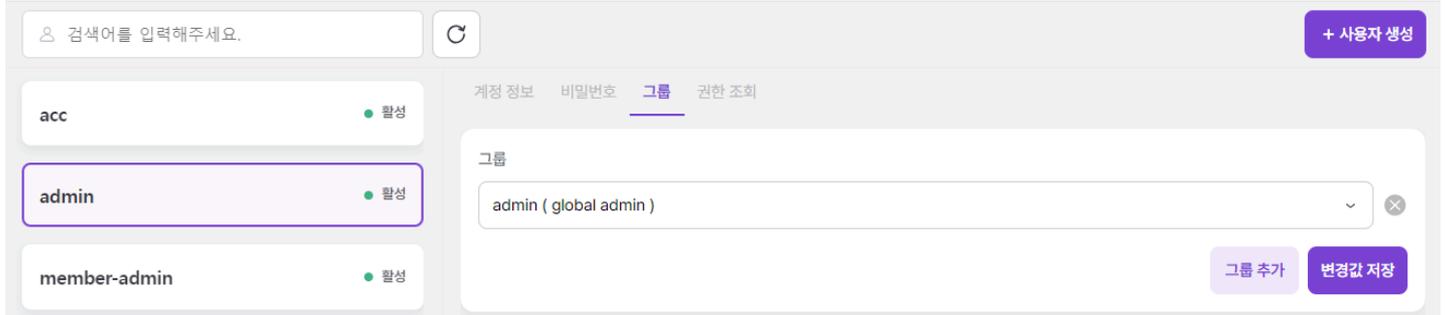
Display Name

그룹을 등록하는 곳에서 실제로 표시될 이름을 정의한다.

NOTE

사용자를 그룹에 연결하는 것은 사용자 메뉴에서 사용자 정보의 그룹 정보를 변경해서 수행한다.

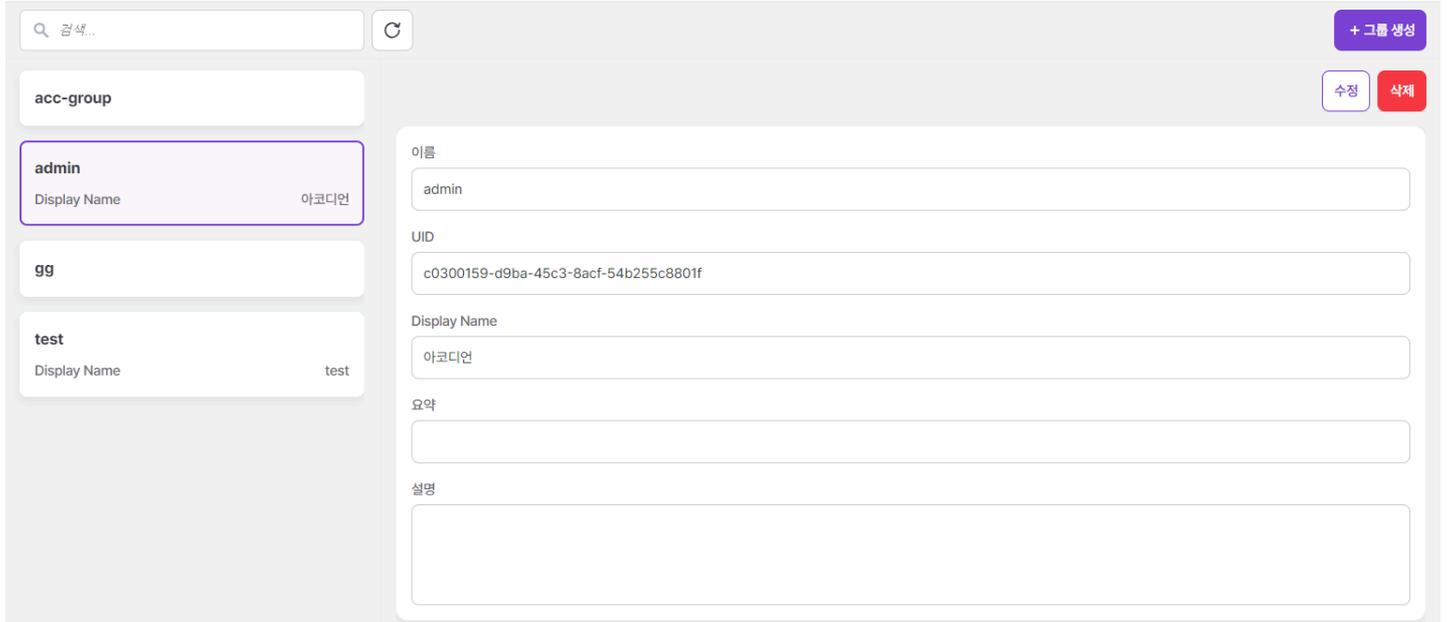
계정 > 사용자



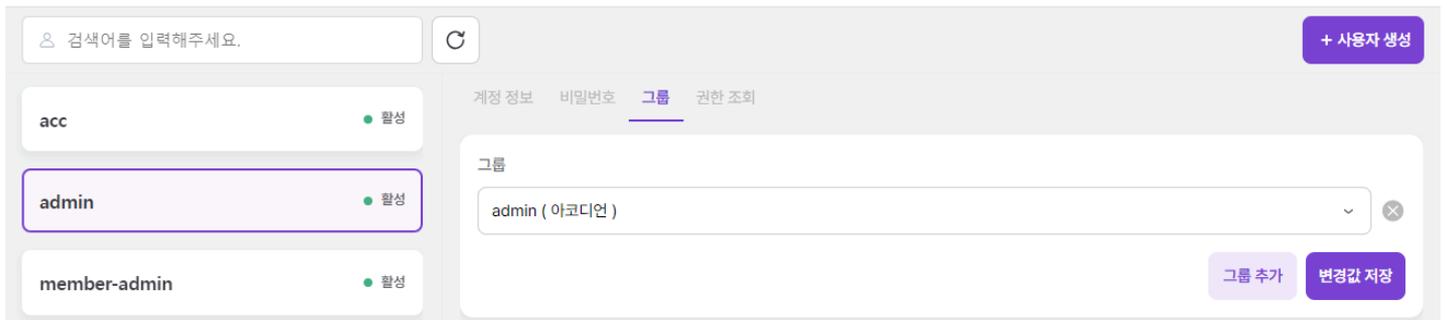
위의 이미지와 같이 그룹명 (Display Name) 으로 출력된다.

등록된 그룹의 정보를 수정하여 Display Name 에 정보를 수정하면 해당 내용으로 출력된다.

계정 > 그룹

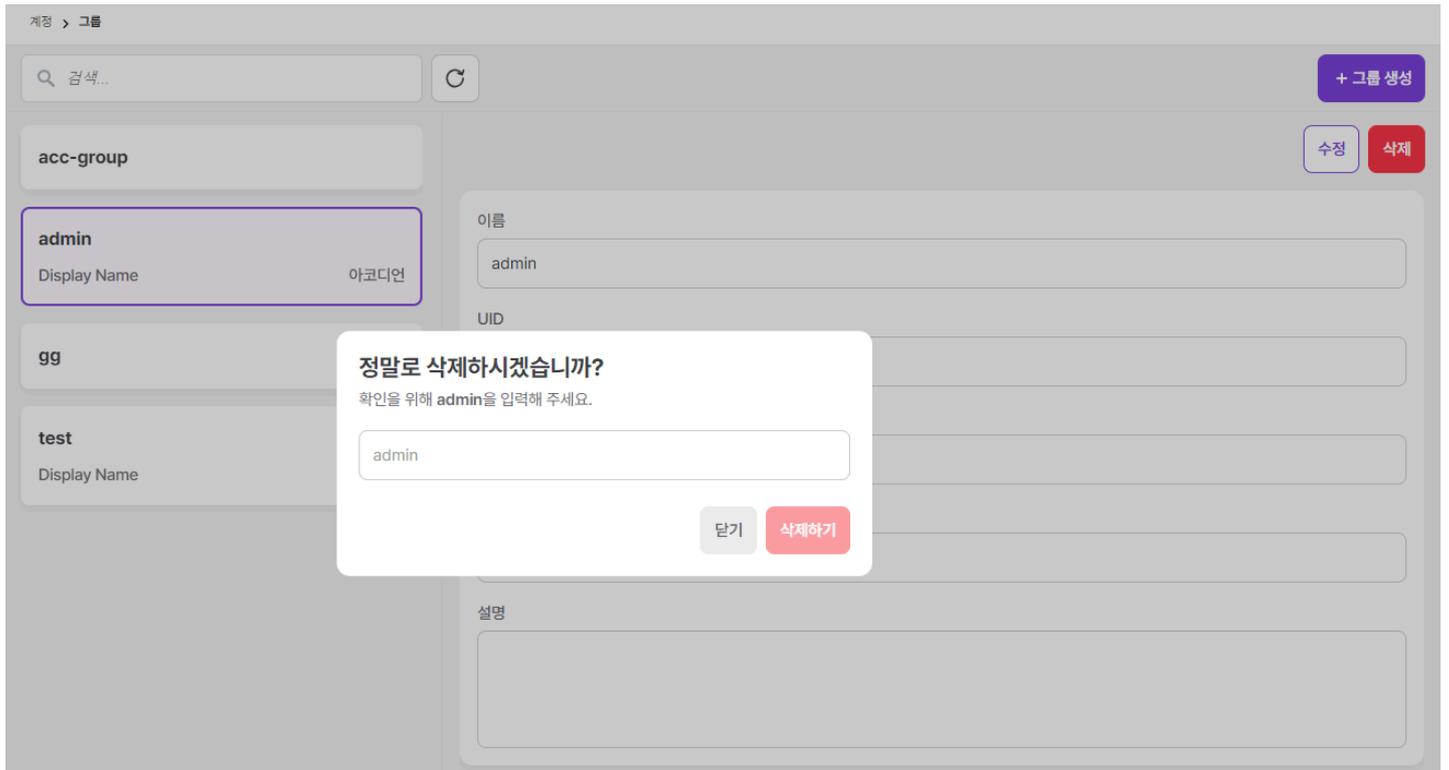


계정 > 사용자



그룹 삭제

그룹을 삭제하기 위해서는 목록에서 삭제할 대상이 되는 그룹을 선택하고 우측 상단의 삭제 버튼을 선택한다. 이후 나타나는 모달에서 그룹 이름을 입력하고 삭제하기 버튼을 선택하면 완료된다.



4.1.4.3. 권한

권한은 여러 메뉴와 룰의 집합을 의미한다. 이러한 권한은 스코프 별로 따로 관리되며 각각 글로벌/클러스터/네임스페이스 권한이라고 한다. 각 스코프 권한에서는 해당 스코프 이하의 메뉴에 대해 바인딩할 수 있다. 예를 들면 아래와 같다.

- 글로벌 권한
 - 클러스터 메뉴에 대한 룰 설정 가능 (글로벌 스코프)
 - 네임스페이스 메뉴에 대한 룰 설정 가능 (클러스터 스코프)
 - 카탈로그 메뉴에 대한 룰 설정 가능 (네임스페이스 스코프)
- 클러스터 권한
 - 네임스페이스 메뉴에 대한 룰 설정 가능 (클러스터 스코프)
 - 카탈로그 메뉴에 대한 룰 설정 가능 (네임스페이스 스코프)
- 네임스페이스 권한
 - 카탈로그 메뉴에 대한 룰 설정 가능 (네임스페이스 스코프)

TIP

본문에서는 글로벌 권한을 예로 설명한다.

글로벌/클러스터/네임스페이스 권한 메뉴에서 각 스코프별 권한 정보를 조회할 수 있다. 하나의 권한은 하나 이상의 메뉴와 룰로 구성된다.

계정 > 글로벌 권한

+ 글로벌 권한 생성

global-admin
2024. 6. 20. 오후 5:35:12

global-viewer
2024. 6. 20. 오후 5:35:12

이름

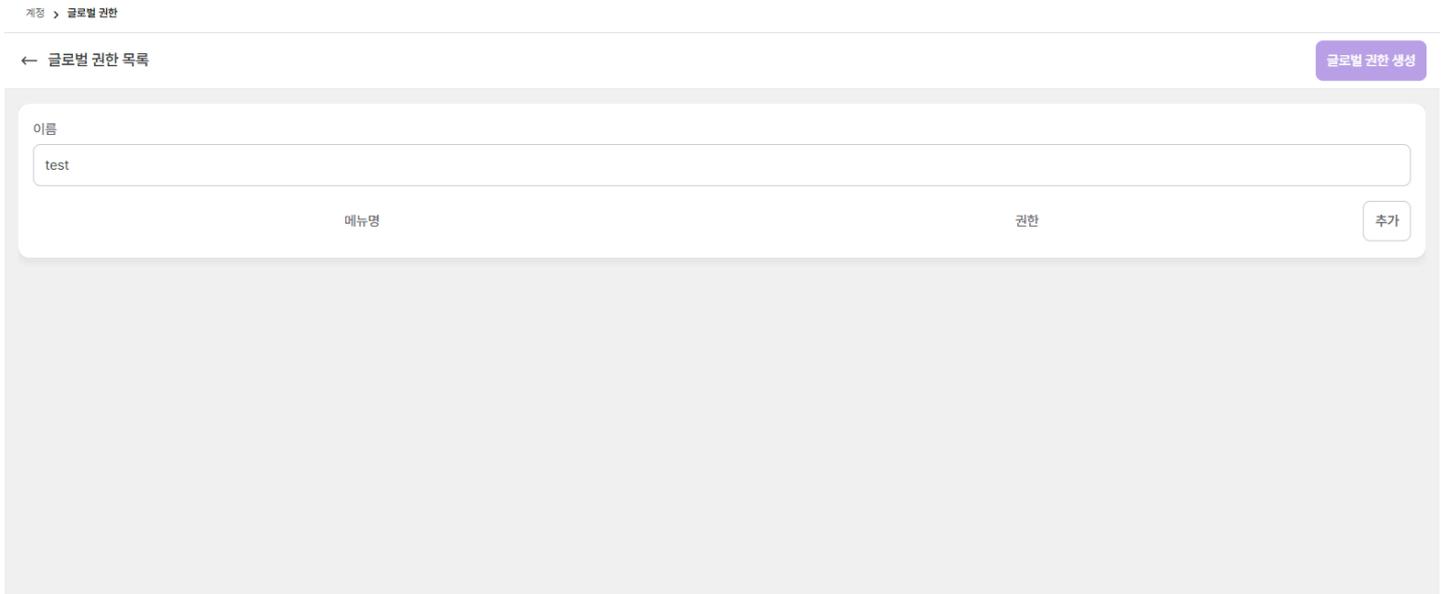
메뉴명	권한
글로벌 대시보드	viewer
클러스터	admin
클러스터 대시보드	viewer
네임스페이스	admin
노드	admin
네임스페이스 대시보드	viewer
헬름	admin
사용자	admin
그룹	admin
글로벌 권한	admin
클러스터 권한	admin
네임스페이스 권한	admin
글로벌 멤버	admin

수정
삭제

> 이벤트 (NORMAL 0 / WARNING 0)

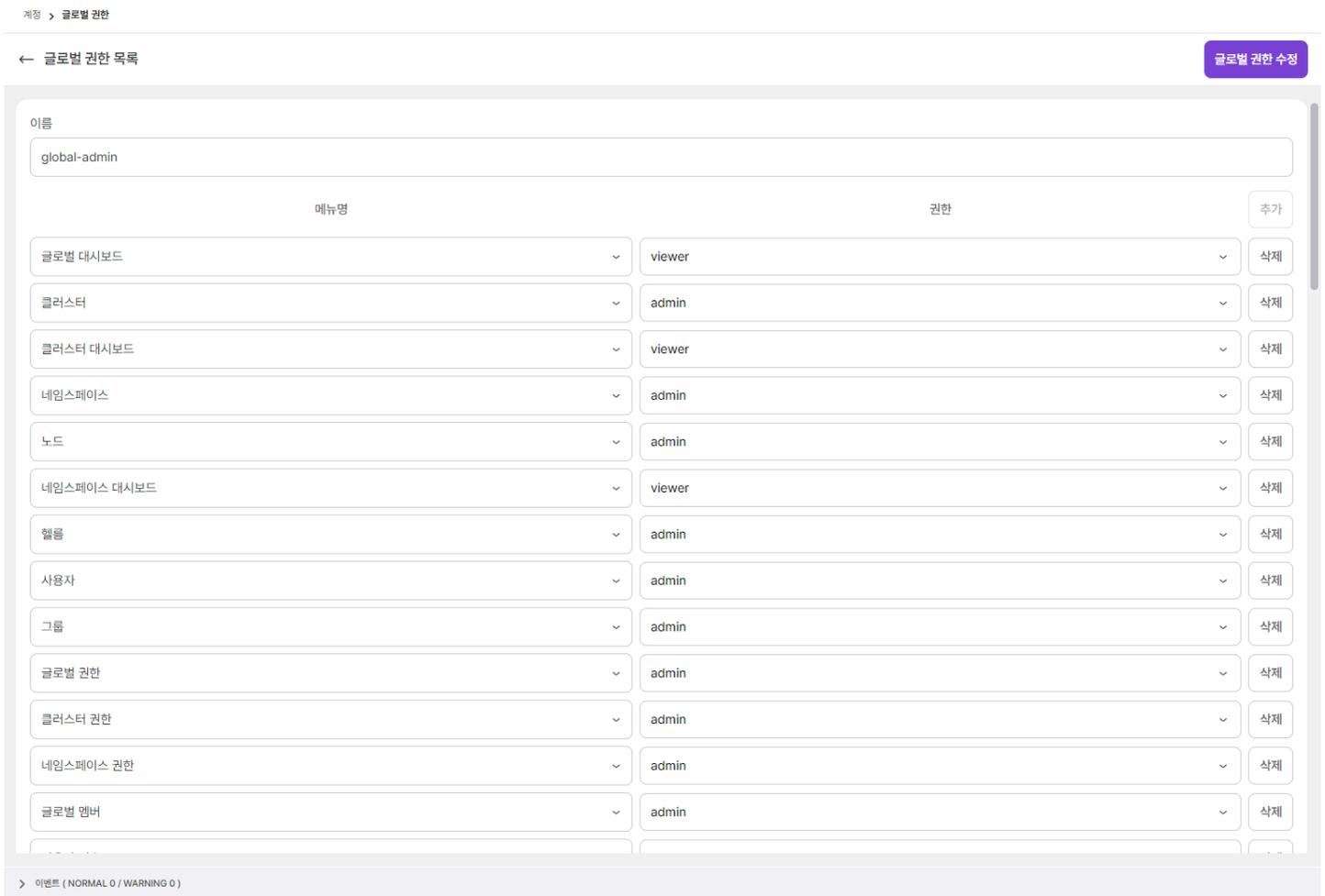
권한 등록

권한을 등록하기 위해서는 우측 상단의 + 글로벌 권한 생성 버튼을 선택하고 권한의 이름과 권한에 포함할 메뉴 및 롤을 추가한다.



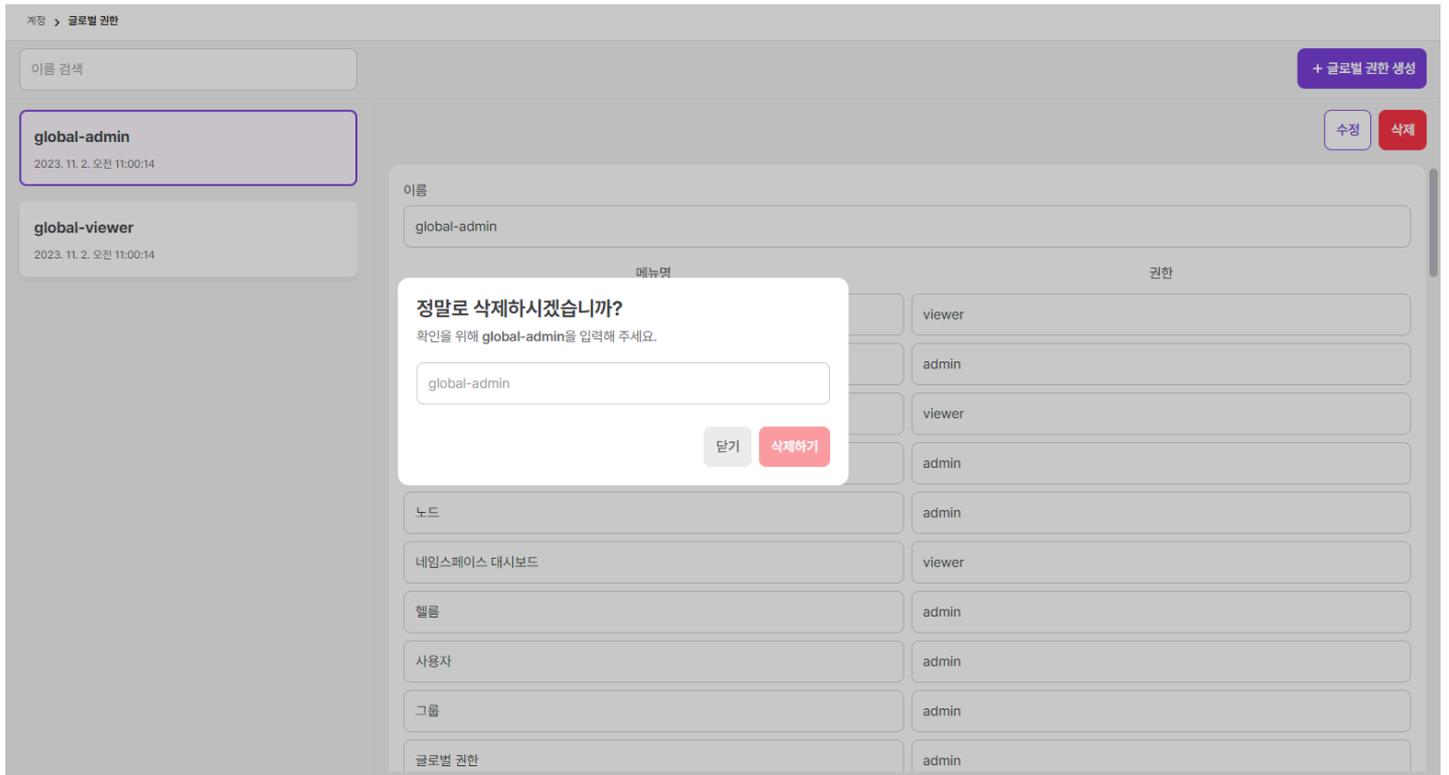
권한 수정

권한을 수정하기 위해서는 수정하려는 권한을 선택하고 우측 상단의 수정 버튼을 선택하여 수정 화면으로 이동한다. 그리고 메뉴를 추가하거나 삭제, 또는 메뉴에 대한 롤을 변경한다. 마지막으로 우측 상단의 글로벌 권한 수정 버튼을 선택하여 변경 사항을 권한에 반영한다.



권한 삭제

권한을 삭제하기 위해서는 삭제하려는 권한을 선택하고 우측 상단의 삭제 버튼을 선택한다. 이후 표시되는 모달에서 권한 이름을 입력하고 삭제하기 버튼을 선택한다.



4.1.4.4. 글로벌 멤버

글로벌 멤버는 사용자와 그룹의 글로벌 권한을 관리한다.

계정 > 글로벌 멤버

저장

사용자 추가

이름 ↑	권한	
사용자 admin	권한 global-admin	수정 삭제

이전 1 다음

그룹 추가

이름 ↑	권한	
그룹 acc-group	권한 global-admin	수정 삭제

이전 1 다음

종류	설명
사용자	개별 사용자의 글로벌 권한 관리
그룹	그룹 별 글로벌 권한 관리 (그룹에 있는 사용자들에게 모두 권한을 부여하는 것과 같은 효과)

멤버 추가

사용자(또는 그룹) 목록 우측의 추가 버튼을 선택한다. 모달에서 사용자(또는 그룹)을 선택해 권한을 설정할 수 있다. 설정 후에는 저장 버튼을 선택해야 변경 사항이 반영된다. 사용자(또는 그룹)은 다중 선택이 가능하지만 권한은 단일 선택만이 가능하다.

계정 > 글로벌 멤버

저장

사용자 추가

이름 ↑	권한	
사용자 admin		수정 삭제

그룹 추가

이름 ↑	권한	
그룹 acc-group	권한 global-admin	수정 삭제

이전 1 다음

사용자

사용자 추가

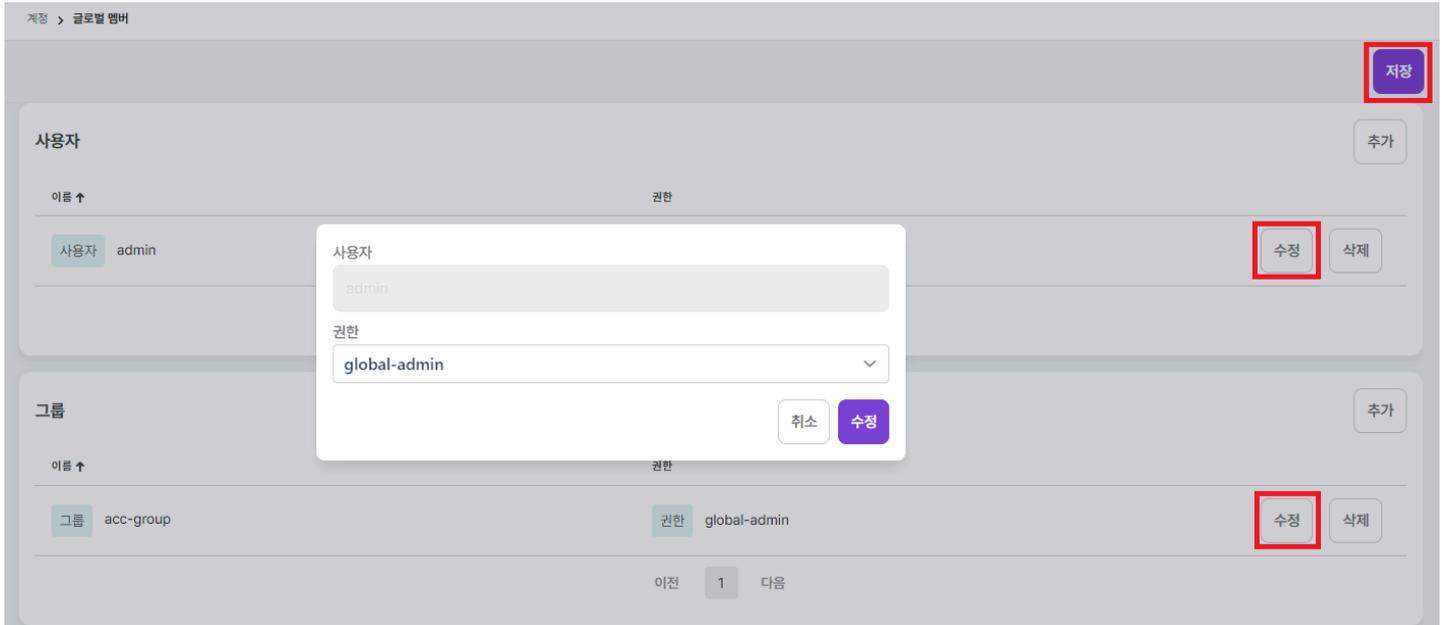
권한

권한 추가

취소 **추가**

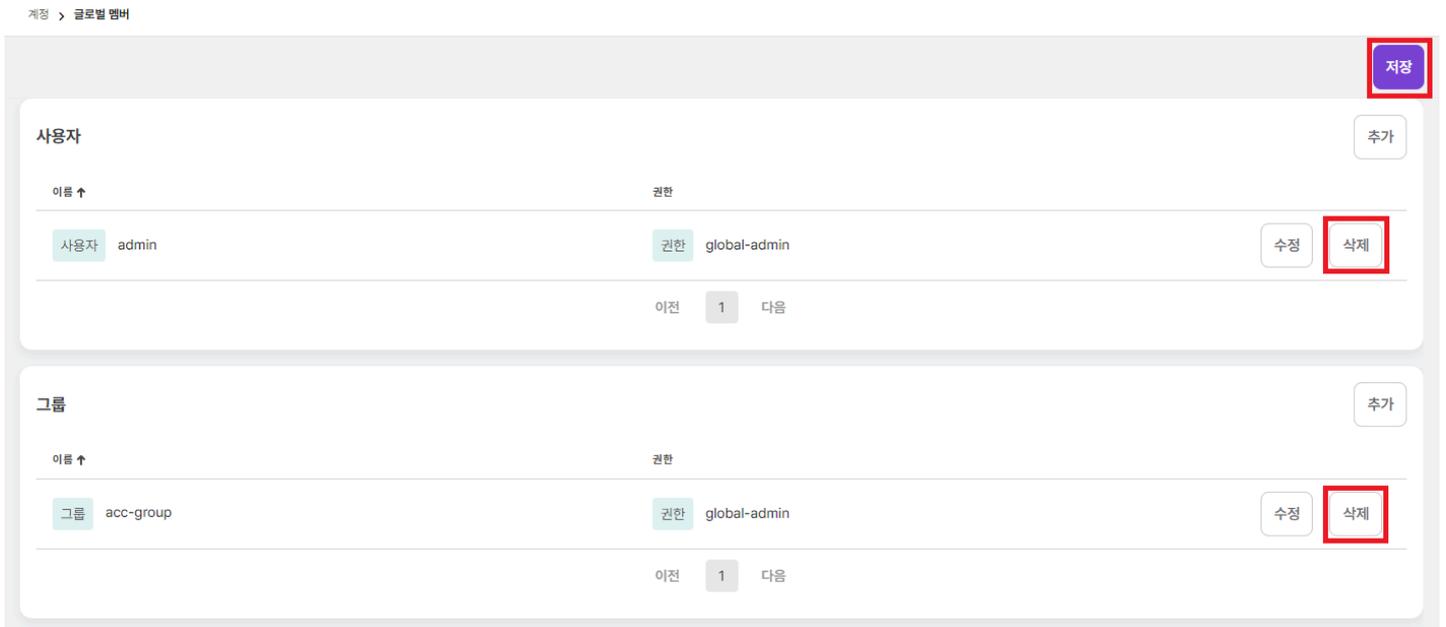
멤버 권한 수정

사용자(또는 그룹) 우측의 수정 버튼으로 해당 사용자(또는 그룹)의 권한을 수정할 수 있다. 이후 상단의 저장 버튼을 클릭해야만 변경 사항이 반영된다.



멤버 삭제

사용자(또는 그룹)의 우측 삭제 버튼을 선택하면 멤버를 삭제할 수 있다. 이후 상단의 저장 버튼을 클릭해야만 변경 사항이 반영된다.



4.1.4.5. 사용자 접속 로그

사용자들의 로그인/로그아웃과 같은 접속 로그 정보를 조회할 수 있다.

계정 > 사용자 접속 로그

타입	사용자명	날짜				
로그인 x 로그아웃 x	사용자 이름을 입력하세요.	2024-09-04 - 2024-09-05	조회			
시간	타입	인증 타입	사용자명	UID	세션 ID	IP 주소
2024. 9. 5. 오후 3:24:18	login	password	admin	57ec3a75-fdd1-4919-8267-006360697ac8	39c3358a-c4ec-4f9b-a015-01a1380eb90c	10.10.178.50
2024. 9. 5. 오후 3:24:18	login	password	admin	57ec3a75-fdd1-4919-8267-006360697ac8	1ecaf7e-e83d-4b86-8a46-689e473ea68c	10.10.178.50
2024. 9. 5. 오후 3:24:18	login	password	admin	57ec3a75-fdd1-4919-8267-006360697ac8	0138607a-0b26-4dcd-9784-c3d0fe3de741	10.10.178.50
2024. 9. 5. 오후 3:24:18	login	password	admin	57ec3a75-fdd1-4919-8267-006360697ac8	82aec4a9-8a7a-4ad6-94cc-e92c0b309e84	10.10.178.50
2024. 9. 5. 오후 3:24:18	login	password	admin	57ec3a75-fdd1-4919-8267-006360697ac8	710ebf70-7161-4bad-88c7-9d63c1e78ec2	10.10.178.50
2024. 9. 5. 오후 3:24:18	login	password	admin	57ec3a75-fdd1-4919-8267-006360697ac8	a0fc5654-2fa5-411b-8075-1f731d4da4cc	10.10.178.50
2024. 9. 5. 오후 3:24:18	login	password	admin	57ec3a75-fdd1-4919-8267-006360697ac8	dd0bfb3f-b361-481a-9ff9-1c0fafd638c1	10.10.178.50
2024. 9. 5. 오후 3:24:10	login	password	dhcho-viewer	f5e92e25-2ab2-4184-9feb-09c7a51298d0	d763cc74-e7f9-44e8-8dc9-a98cfe8e4e6	10.10.255.250
2024. 9. 5. 오후 3:24:10	login	password	admin	57ec3a75-fdd1-4919-8267-006360697ac8	4e6e2670-c67e-4cf8-8bfa-e545a19ee06d	10.10.178.50
2024. 9. 5. 오후 3:24:10	login	password	admin	57ec3a75-fdd1-4919-8267-006360697ac8	e5a3e47d-dbca-425a-9ab0-b070787d61f4	10.10.178.50
2024. 9. 5. 오후 3:24:09	login	password	admin	57ec3a75-fdd1-4919-8267-006360697ac8	c947ce5d-67ce-4b90-a384-e40d3fe20e36	10.10.178.50
2024. 9. 5. 오후 3:23:35	login	password	dhcho-viewer	f5e92e25-2ab2-4184-9feb-09c7a51298d0	d62ebaa8-1f2a-4df2-981e-f7c39559d368	10.10.255.250
2024. 9. 5. 오후 3:23:26	login	password	dhcho-viewer	f5e92e25-2ab2-4184-9feb-09c7a51298d0	df1fac38-b247-4898-9fd2-072af9efe7c8	10.10.255.250
2024. 9. 5. 오후 3:23:26	login	password	dhcho-viewer	f5e92e25-2ab2-4184-9feb-09c7a51298d0	a099f25a-5b3e-46fd-b90a-e4669ae0f868	10.10.255.250
2024. 9. 5. 오후 3:23:26	login	password	dhcho-viewer	f5e92e25-2ab2-4184-9feb-09c7a51298d0	ab805f15-8de2-434c-86fc-	10.10.255.250

항목	설명
시간	시간
타입	접속 타입 (LOGIN 또는 LOGOUT)
인증 타입	승인 타입(인증 방식)으로 아코디언 웹 콘솔에서 로그인인 아닌 Basic 인증한 경우 password 인증 타입을 가진다.
사용자명	사용자 명
UID	사용자 UID
세션 ID	사용자 세션 ID
IP 주소	사용자 IP 주소

상단의 조건을 설정하여 보다 구체적으로 검색을 할 수 있다. 조건을 설정하고 [조회](#) 를 선택하면 해당 조건으로 사용자 접속 로그를 검색한다. 설정 가능한 조건은 다음과 같다.

항목	설명
타입	타입으로 필터링 (다중 선택 가능)
사용자명	사용자 명으로 필터링
시간	캘린더로 날짜 범위를 지정

TIP

조회 후 테이블 헤더에 조건을 설정하면 결과 내 검색이 가능하다.

4.1.5. 글로벌 설정

글로벌 설정에서는 시스템 전반에 영향을 미치는 활성화 키와 알림 발송을 받는 수신자에 대한 설정이 가능하다.

4.1.5.1. 활성화 키

아코디언을 사용하기 위한 활성화 키를 관리한다.

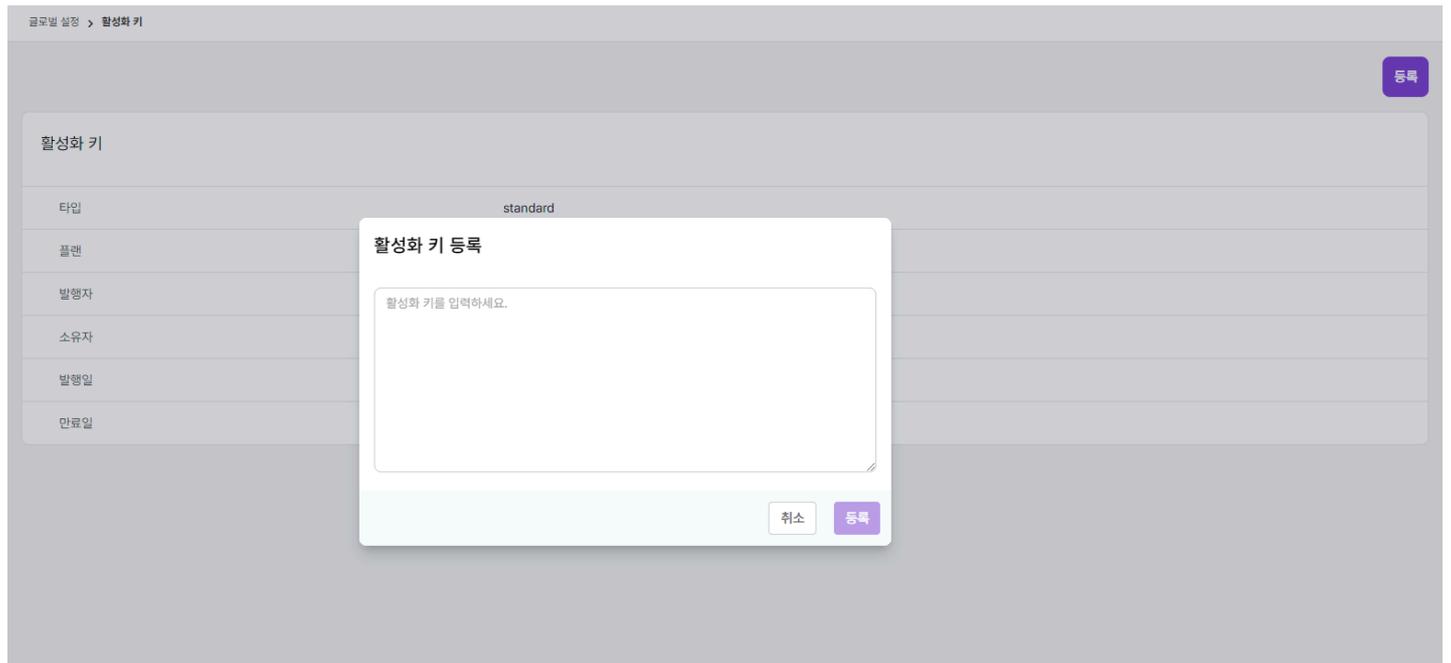
글로벌 설정 > 활성화 키

활성화 키	
타입	standard
플랜	subscription
발행자	Mantech
소유자	dev
발행일	2024-01-02T03:00:00Z
만료일	2024-12-31T14:59:59Z

분류	설명
타입	Standard 또는 Trial <ul style="list-style-type: none"> Standard <ul style="list-style-type: none"> 정식 활성화 키에 의해 관리받을 수 있는 타입 Trial <ul style="list-style-type: none"> 일정 기간동안 아코디언을 사용해 볼 수 있는 타입
플랜	Perpetual 또는 Subscription <ul style="list-style-type: none"> Perpetual <ul style="list-style-type: none"> 영구적인 활성화 키 Subscription <ul style="list-style-type: none"> 기간이 있는 활성화 키
발행자	활성화 키 발행인 (Mantech)
소유자	활성화 키 소유자
발행일	활성화 키 발행일
만료일	활성화 키 만료일

활성화 키 등록

우측 상단의 등록 버튼을 선택한다. 모달이 나타나면 활성화 키를 입력하여 등록한다.



4.1.5.2. 글로벌 수신자

알림 정책에 따라 알림을 받을 수신자를 설정한다.

글로벌 수신자 개요

기본
▼

+ 수신자 생성

sample-receiver

🕒 2024-05-22 20:21:16

수정

삭제

MULTICLUSTERRECEIVER **sample-receiver**

이메일

호스트: smtp.sample.com

포트: 999

발신자 이메일: sample@sample.com

인증

방식: BasicAuth

TLS 활성화: false

사용자 이메일: sample@sample.com

사용자 비밀번호: ***** 👁

수신자 이메일 목록

- user1@sample.com
- user2@sample.com

슬랙

토큰: ***** 👁

채널 목록

- channel1
- channel2

Webhook

Webhook URL: https://sample-hook.com

인증 타입: 인증 안함

바디 test : sample1

헤더 content-type : application/json

TLS 인증서 검증: O

Root CA 인증 sample-ca 👁

서

인증서/개인 키 sample-tls 👁

키

키

- 이메일

분류	설명
호스트	SMTP Host 주소
포트	SMTP Port 번호
발신자 이메일	발신 이메일 주소
방식	인증 방식 인증 안함 (익명), Basic Auth
TLS 활성화	TLS 사용 여부
사용자 이메일	SMTP 사용자 이메일
사용자 비밀번호	SMTP 사용자 비밀번호
수신자 이메일 목록	수신자 이메일 정보

- 슬랙

분류	설명
토큰	슬랙 토큰 값
채널 목록	슬랙 채널명

- 웹훅

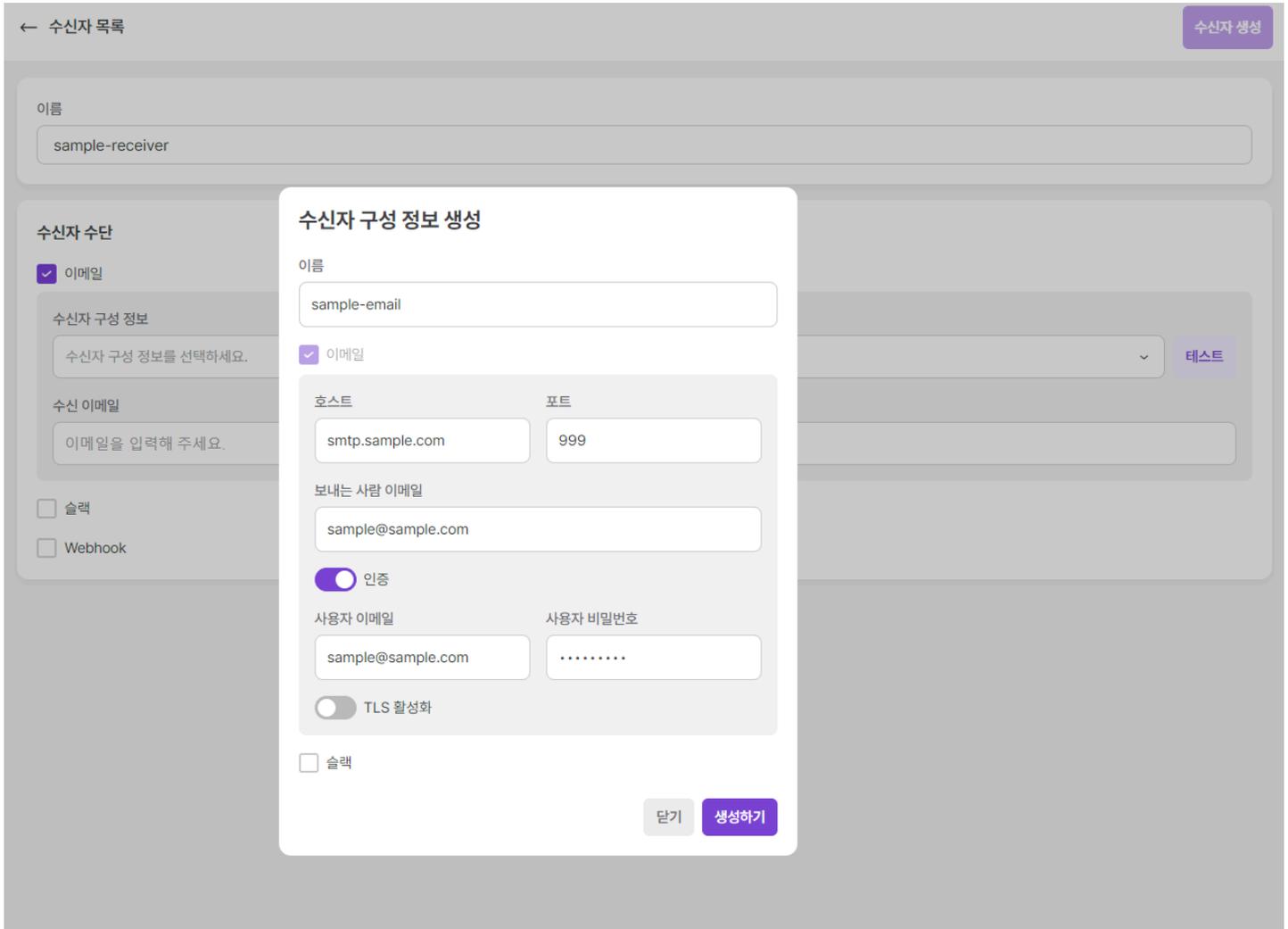
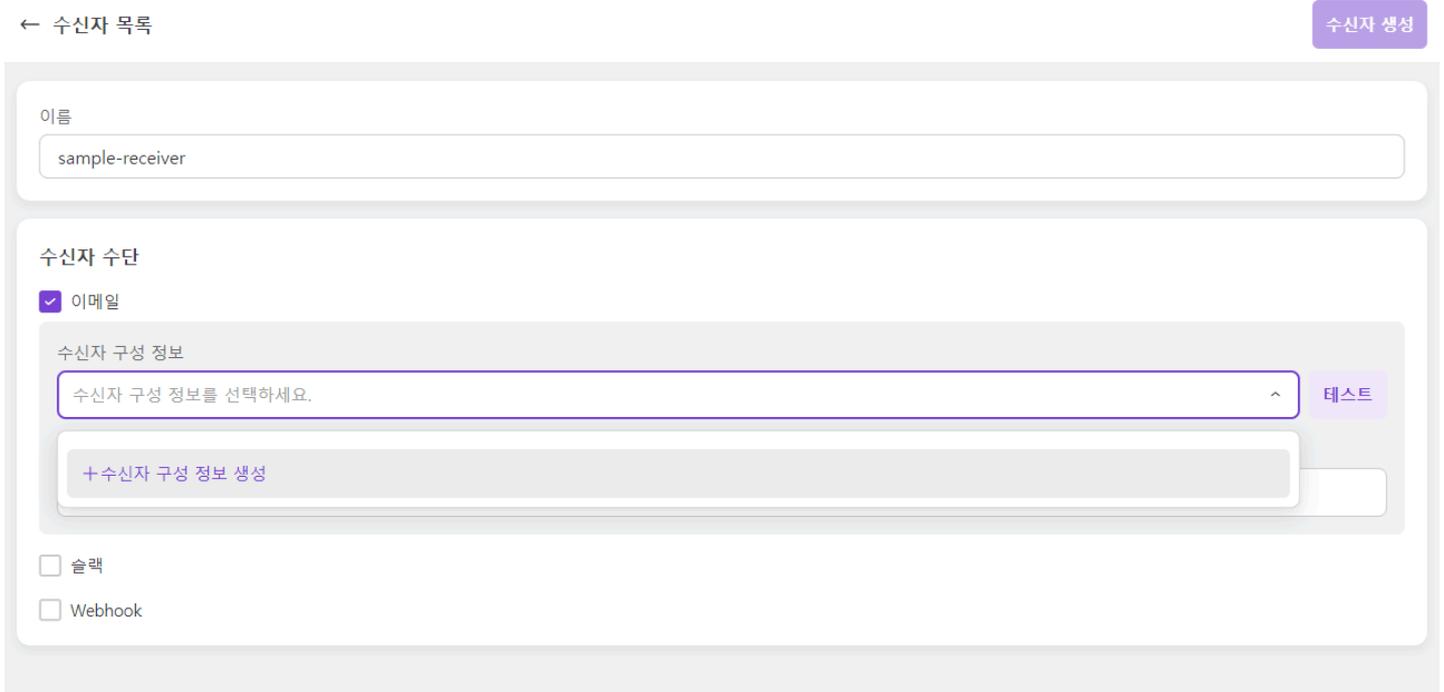
분류	설명
Webhook URL	웹훅 URL 주소
인증 타입	인증 안함 (익명), Bearer 인증, basic 인증
토큰	Bearer 인증 의 경우 토큰 정보
사용자 이름	basic 인증 의 경우 사용자 이름
사용자 비밀번호	basic 인증 의 경우 사용자 비밀번호
바디	요청에 포함될 바디
헤더	요청에 대한 헤더
TLS 인증서 검증	검증 시 0, 미검증 시 X
Root CA 인증서	Root CA 인증서 정보가 담긴 시크릿 이름 및 data
인증서/개인 키	TLS 인증서 및 개인키 정보가 담긴 시크릿 이름 및 data

글로벌 수신자 생성

수신자 생성 시 이메일, 슬랙, 웹훅 을 설정할 수 있다.

이메일 설정

수신자 구성 정보 및 수신 이메일을 입력 후 테스트 버튼을 눌러 유효한 설정인지 확인한다.



← 수신자 목록

수신자 생성

이름

sample-receiver

수신자 수단

이메일

수신자 구성 정보

sample-email

테스트

호스트: smtp.sample.com
포트: 999
발신자 이메일: sample@sample.com

인증

방식: BasicAuth
TLS 활성화: false
사용자 이메일: sample@sample.com
사용자 비밀번호: *****

참조 수신자 목록

수신 이메일

user1@sample.com × user2@sample.com × 이메일을 입력해 주세요.

슬랙

Webhook

슬랙 설정

수신자 구성 정보 및 채널을 입력 후 테스트 버튼을 눌러 유효한 설정인지 확인한다.

← 수신자 목록 수신자 수정

이름
sample-receiver

수신자 수단

이메일

슬랙

수신자 구성 정보

수신자 구성 정보를 선택하세요. 테스트

+수신자 구성 정보 생성

채널을 입력하세요. 채널 추가

Webhook

← 수신자 목록 수신자 수정

이름
sample-receiver

수신자 수단

이메일

슬랙

수신자 구성 정보

수신자 구성 정보를 선택하세요. 테스트

채널

채널을 입력하세요. 채널 추가

Webhook

수신자 구성 정보 생성

이름
sample-slack

이메일

슬랙

토큰
sample-slack-token-string

닫기 생성하기

← 수신자 목록

수신자 생성

이름

sample-receiver

수신자 수단

이메일

슬랙

수신자 구성 정보

sample-slack

테스트

토큰: *****

참조 수신자 목록

채널

channel1

channel2

채널 추가

Webhook

웹훅 설정

웹훅 URL, 헤더, 바디 정보를 입력한다.

인증 안함, Bearer 인증, basic 인증 중 인증 타입을 선택한다.

웹훅 URL, 헤더, 바디 정보 입력 및 인증 타입 선택 후 테스트 버튼을 눌러 유효한 설정인지 확인한다.

← 수신자 목록

수신자 생성

이름

sample-receiver

수신자 수단

이메일

슬랙

Webhook

Webhook url

https://sample-hook.com 테스트

인증 타입

인증 안함 ^

인증 안함

Bearer 인증

basic 인증

바디

{ }

TLS 인증서 검증 건너뛰기

Bearer 인증 선택 시 토큰 정보가 담긴 시크릿 생성 및 선택

토큰 생성

이름

sample-bearer

토큰

sampletoken

닫기

생성하기

basic 인증 선택 시 사용자 및 비밀번호 정보가 담긴 시크릿 생성 및 선택

기본 인증 생성

이름

sample-basic

사용자

sample-admin

비밀번호

.....

닫기

생성하기

TLS 인증서 검증 건너뛰기 해제 시

Root CA 인증서 및 인증서/개인키를 동일한 방법으로 생성 및 선택

← 수신자 목록

수신자 생성

슬랙

Webhook

Webhook url

https://sample-hook.com

테스트

인증 타입

인증 안함

헤더

content-type

application/json

헤더 추가

바디

```
{
  "test": "sample1"
}
```

TLS 인증서 검증 건너뛰기

Root CA 인증서

sample-ca

인증서/개인 키

sample-tls

Body 예시

```
바디
{
  "test": "sample1"
}
```

웹훅 바디는 JSON 형식으로 입력한다. 그러면 응답 관련 리소스인 notification manifest 와 함께 작성한 Json 형식의 데이터가 웹훅 발생 시 함께 전송된다.

```
{
  "test": "sample1",
  "manifest": {
    "kind": "ClusterNotification",
    "apiVersion": "alert.accordions.co.kr/v1beta1",
    "metadata": {
      ...
    },
    ...
  }
}
```

CAUTION

글로벌 수신자 생성 시 생성된 수신자/수신자 구성 정보는 모든 멤버 클러스터로 자동 복사됩니다. 이 수신자/수신자 구성 정보에는 SMTP 서버의 계정과 비밀번호, Slack이나 Webhook 서버의 사용자 계정(사용자명, 비밀번호)과 토큰, 인증서 정보 등의 기밀정보가 포함될 수 있습니다.

기밀정보는 각 멤버 클러스터의 Secret에 보관됩니다. 글로벌 수신자를 사용하기 전에 다음 사항을 반드시 숙지하세요.

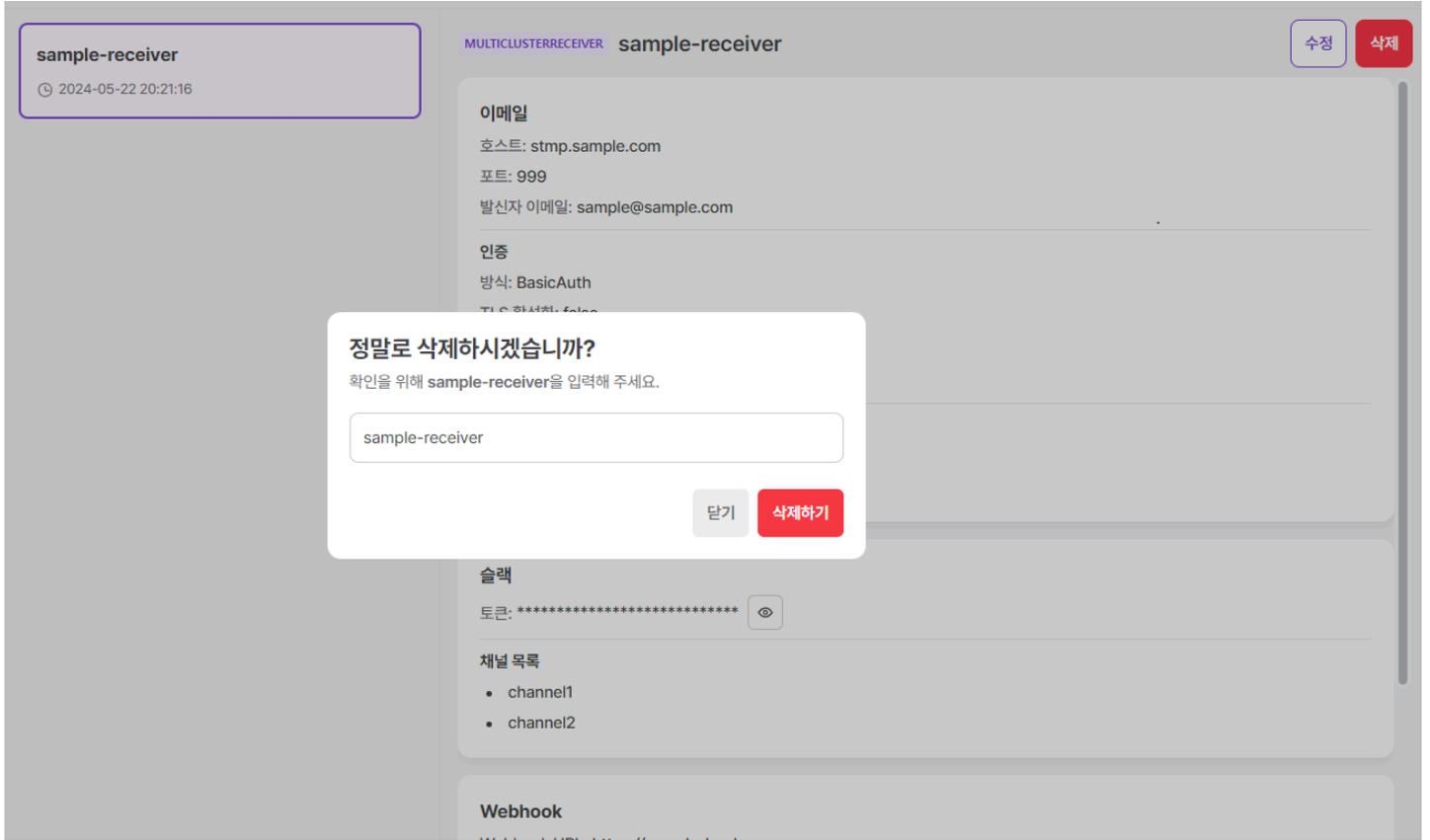
- 정보 유출 위험: 복사된 수신자/수신자 구성 정보는 모든 멤버 클러스터에 배포되므로, 각 클러스터의 보안 수준이 매우 중요합니다. 클러스터에 접근 권한을 철저히 관리하여 무단 접근을 방지하세요. 클러스터 수신자/수신자 구성 정보 중 기밀정보는 acc-system 네임스페이스의 secret으로 관리됩니다. 클러스터 관리자의 권한 사용자는 기밀정보의 접근이 가능하나, 그 외 네임스페이스 접근 사용자는 조회를 할 수 없습니다. 자세한 내용은 [Kubernetes의 RBAC\(역할 기반 접근 제어\)](#) (<https://kubernetes.io/docs/reference/access-authn-authz/rbac/>)를 참고하세요.

글로벌 수신자 수정

수신자 생성과 동일하게 이메일, 슬랙, 웹훅 내용 변경 후 테스트 버튼을 각각 눌러 유효한 설정인지 확인한 후 수정 버튼을 선택하여 반영한다.

글로벌 수신자 삭제

삭제하려는 수신자를 선택하고 우측의 삭제 버튼을 선택한다.

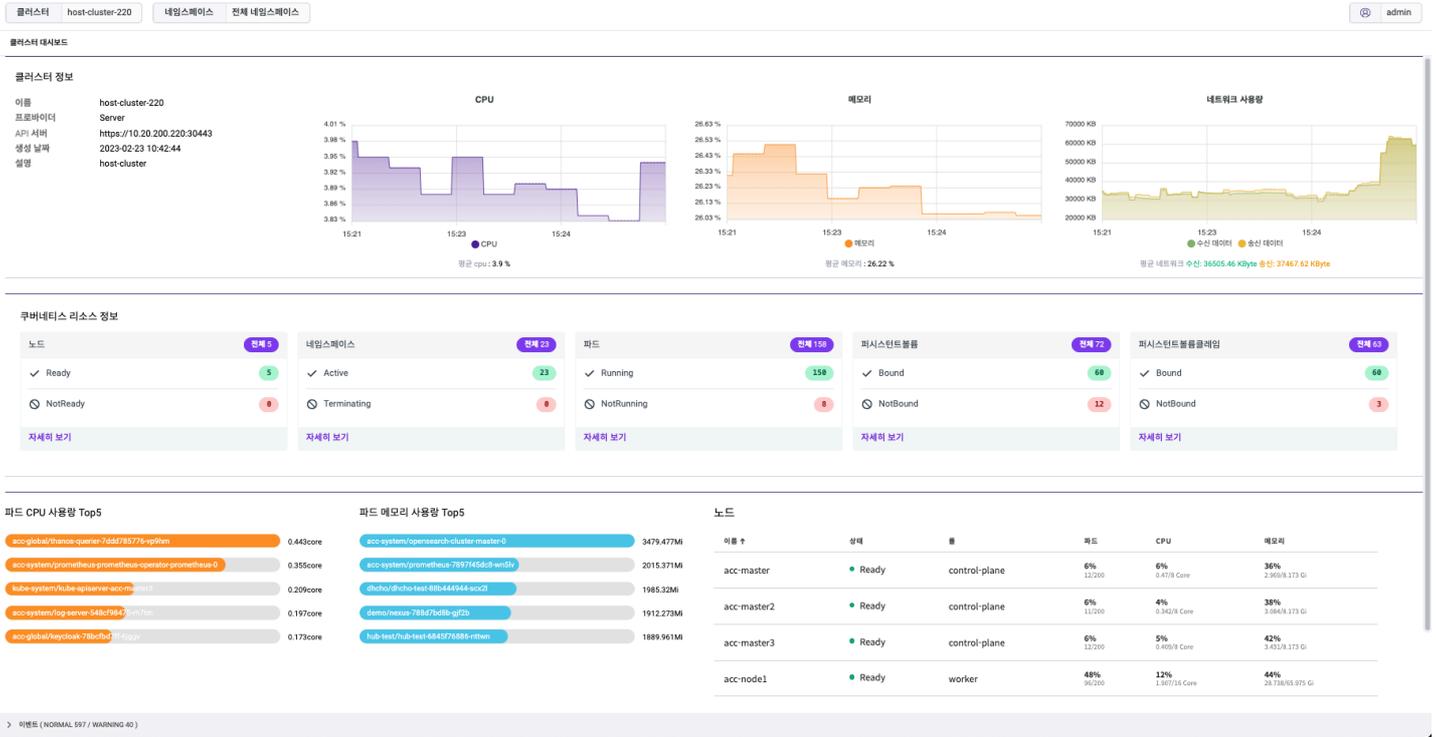


모달에서 수신자 이름을 입력 후 삭제하기 버튼을 선택한다.

4.2. 클러스터 메뉴

4.2.1. 클러스터 대시보드

클러스터 대시보드는 클러스터의 쿠버네티스 리소스 및 CPU/메모리와 같은 시스템 리소스 사용 현황 정보를 제공한다.



표시되는 정보는 다음과 같다.

항목	설명
클러스터 정보	<p>클러스터 기본 정보 및 CPU, 메모리, 네트워크 사용 정보를 제공한다.</p> <ul style="list-style-type: none"> 이름: 클러스터 이름 프로바이더: 쿠버네티스 제공자 API 서버: 아코디언 멤버 에이전트 엔드포인트 생성 날짜: 클러스터 등록 시간 설명: 클러스터 설명 <p>NOTE 생성 날짜 시간 기준은 UTC 이다.</p>
쿠버네티스 리소스 정보	<p>클러스터에 배포된 쿠버네티스 리소스 정보를 제공한다. 자세히 보기 클릭 시 해당 리소스의 목록 화면으로 이동하며 상세 정보를 확인할 수 있다.</p> <p>NOTE 파드의 경우 상태를 클릭하면 해당 상태로 필터링된 파드 목록 화면으로 이동한다.</p>

항목	설명
파드 CPU/메모리 사용량 Top5	클러스터에 배포된 파드 중 CPU/메모리 사용량이 많은 순서대로 5개의 파드 정보를 제공한다.
노드	<p>클러스터를 구성하는 노드들의 정보를 제공한다.</p> <ul style="list-style-type: none"> • 이름: 노드 이름 • 상태: 노드 상태 (Ready 또는 NotReady) • 룰: 노드 역할 (control-plane 또는 worker) • 파드: 노드에 배포된 파드 현황 (배포된 파드의 수/최대 배포가능한 파드의 수) • CPU: 노드 CPU 사용 현황 (사용중인 코어 수/노드의 코어 수) • 메모리: 노드 메모리 사용 현황 (사용중인 메모리 용량/노드의 메모리 용량)

4.2.2. 네임스페이스

네임스페이스는 클러스터에서 쿠버네티스 리소스 그룹을 격리하는 역할을 수행한다. 네임스페이스 기반의 리소스의 경우 리소스의 이름이 네임스페이스 내에서 유일해야 한다. 이를 이용해 팀 또는 프로젝트 별로 네임스페이스를 구성하여 논리적으로 독립시킬 수 있다.

네임스페이스 메뉴에서는 클러스터 내 네임스페이스를 관리할 수 있다. 네임스페이스의 생성, 수정, 삭제가 가능하며 CPU나 메모리와 같은 시스템 리소스에 대해 사용 제한 등을 설정할 수 있으며 제공하는 정보는 다음과 같다.

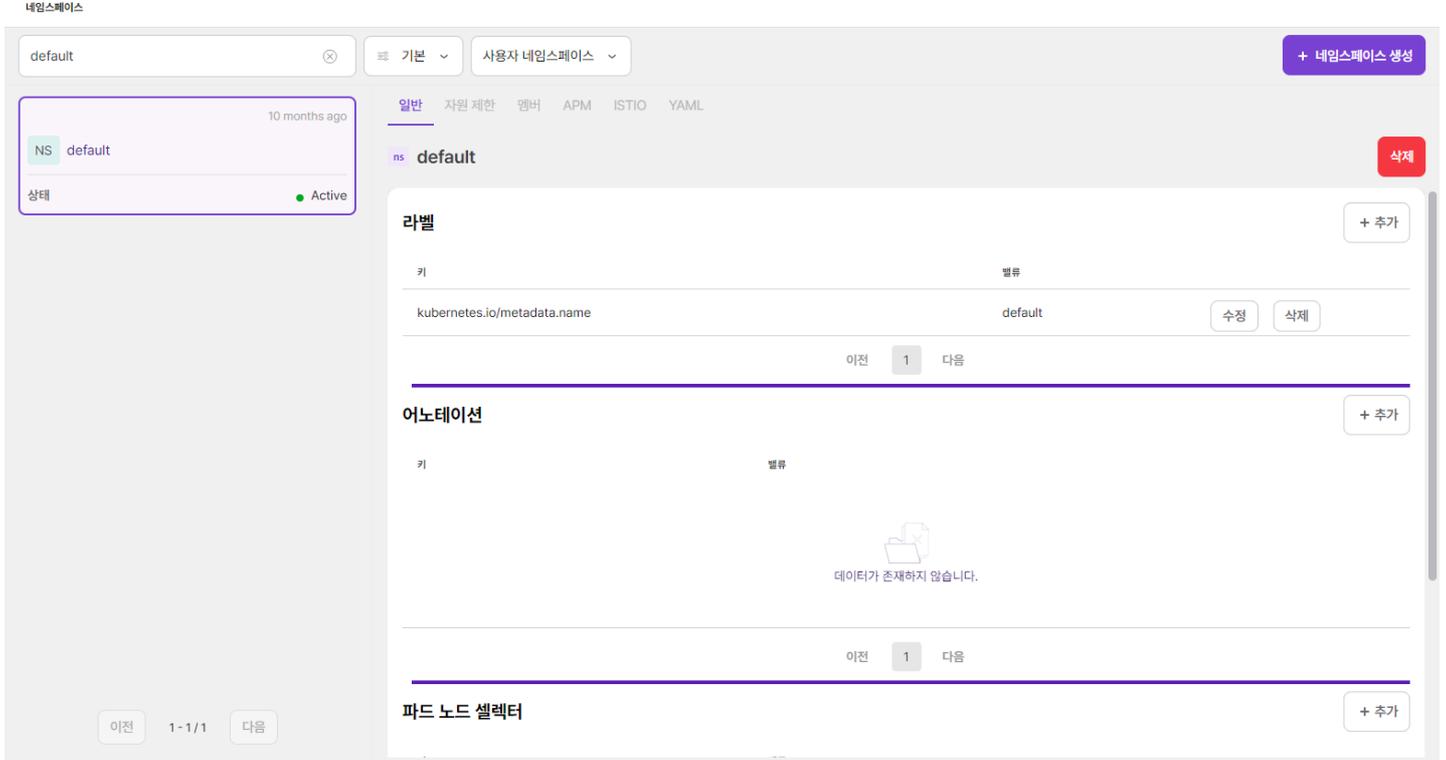


Table 1. 네임스페이스 목록

항목	설명
사용자 네임스페이스	사용자가 생성한 네임스페이스로 애플리케이션을 배포
시스템 네임스페이스	쿠버네티스 및 아코디언 운영에 필요한 리소스가 배포된 네임스페이스

Table 2. 일반

항목	설명
라벨	네임스페이스 레이블로 조회 시 레이블을 이용해 필터링 가능
어노테이션	네임스페이스 어노테이션으로 네임스페이스 설정 등을 저장
파드 노드 셀렉터	파드 배포 시 해당 레이블의 노드에 배포하도록 설정

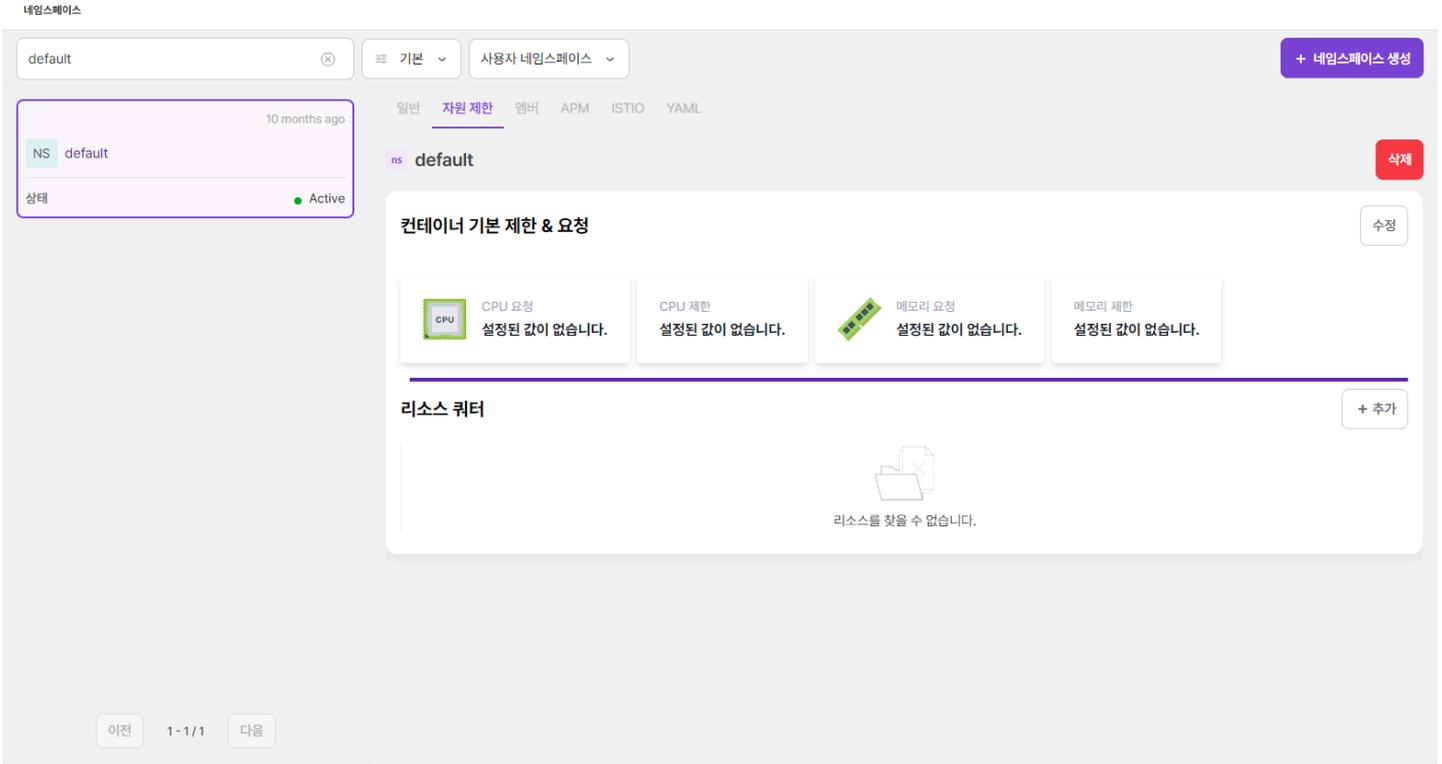


Table 3. 자원 제한

항목	설명
컨테이너 기본 제한 & 요청	쿠버네티스의 리밋레인지어를 이용해 시스템 리소스(CPU/메모리) 제한 설정
리소스 쿼터	쿠버네티스의 리소스쿼터를 이용해 쿠버네티스 리소스 및 시스템 리소스(CPU/메모리) 제한 설정

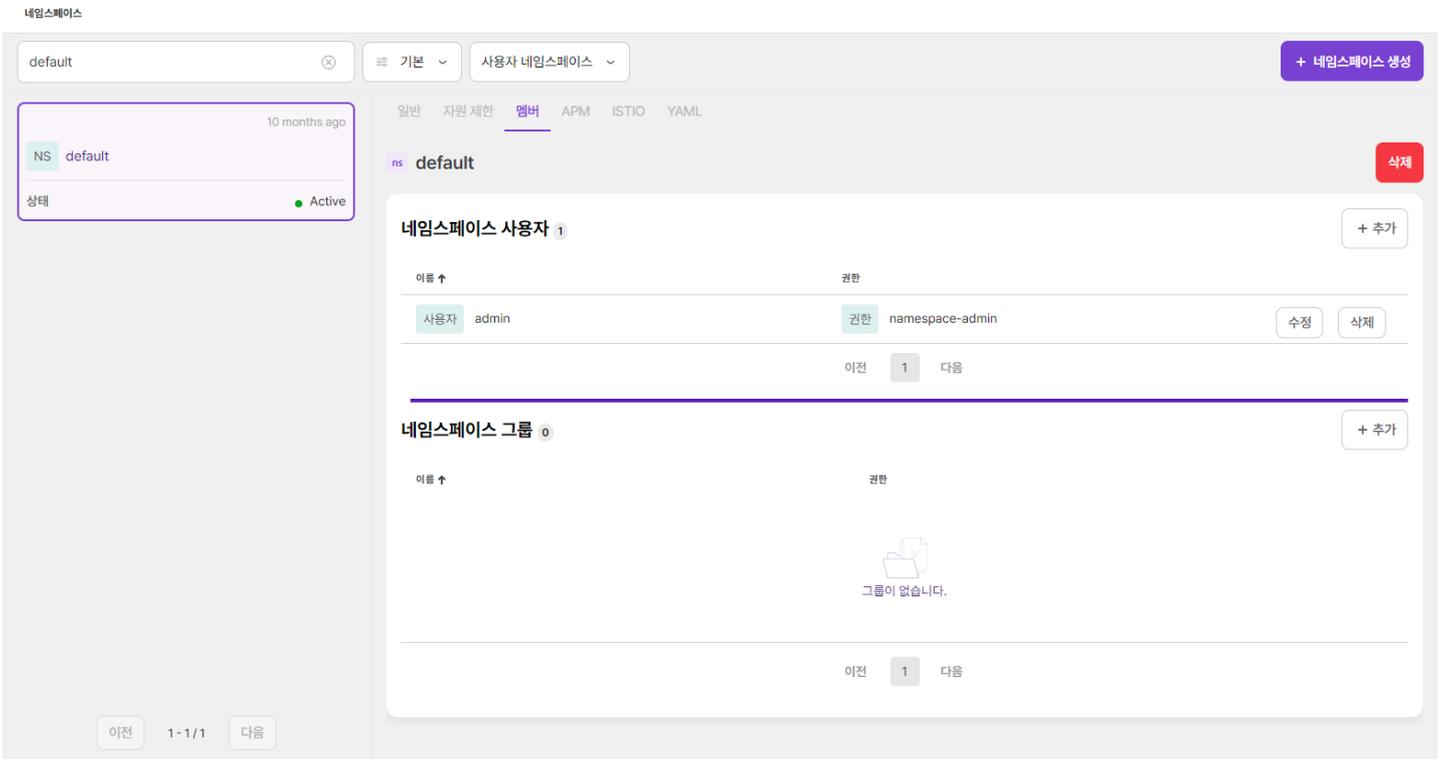


Table 4. 멤버

항목	설명
----	----

항목	설명
네임스페이스 사용자	네임스페이스 사용자 할당 및 권한 설정
네임스페이스 그룹	네임스페이스 사용자 그룹 할당 및 권한 설정

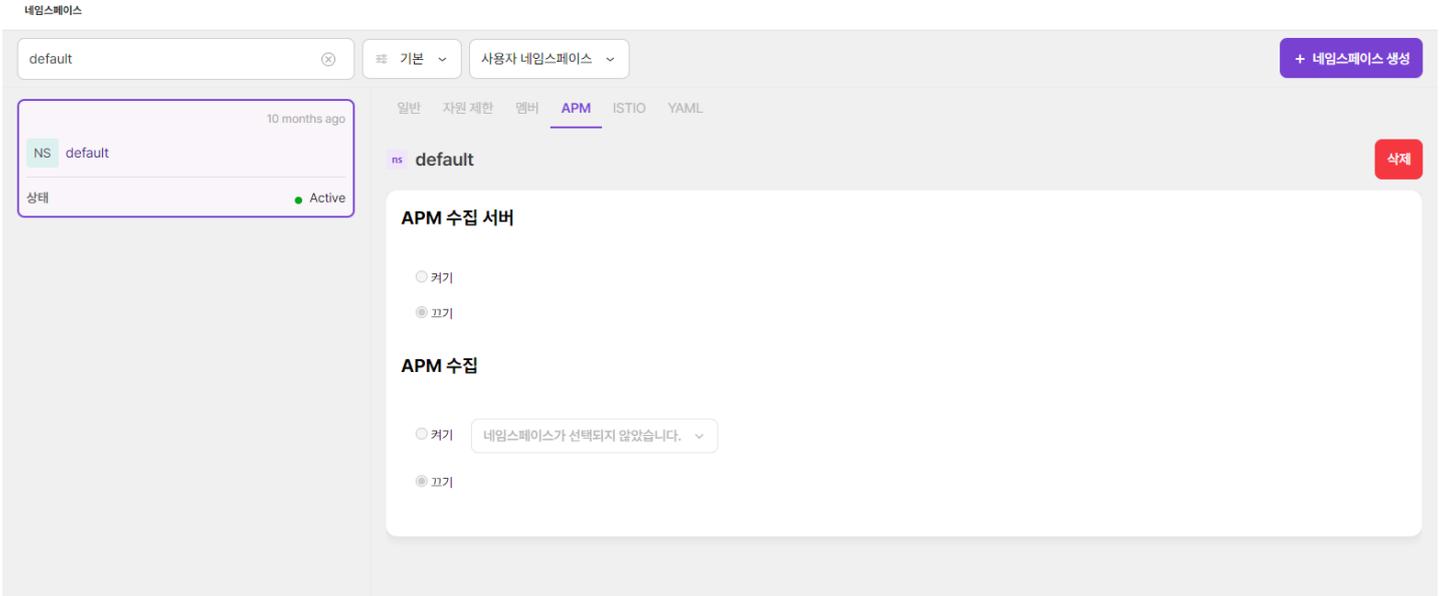


Table 5. APM

항목	설명
APM 수집 서버	네임스페이스 내 스카우터 서버 배포 설정
APM 수집	네임스페이스 내 스카우터 에이전트의 타겟 수집 서버 설정 (타겟 수집 서버의 네임스페이스를 선택)

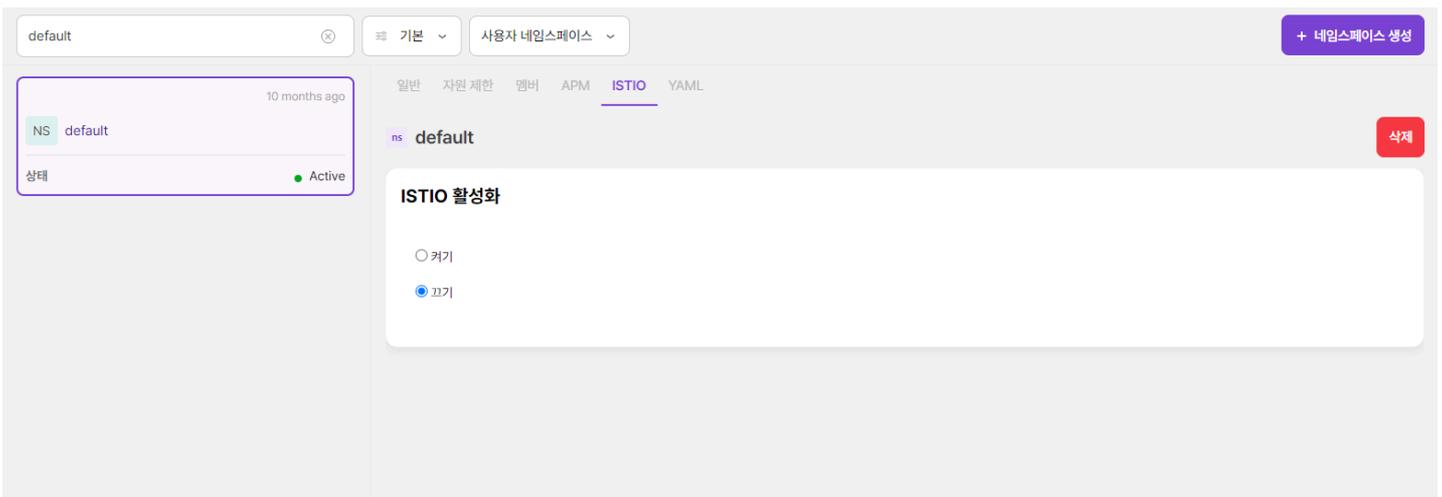


Table 6. ISTIO

항목	설명
ISTIO 활성화	istio-injection 수행 설정

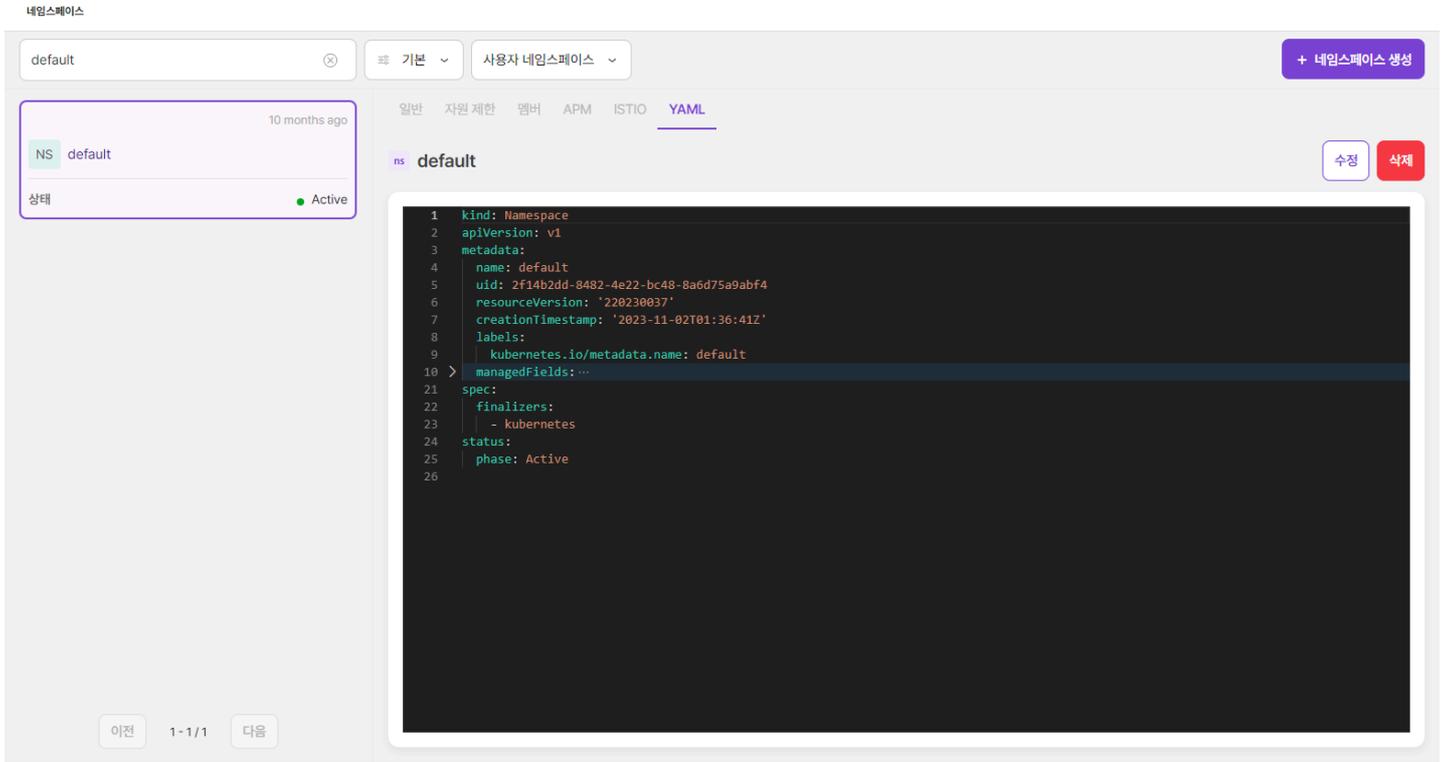
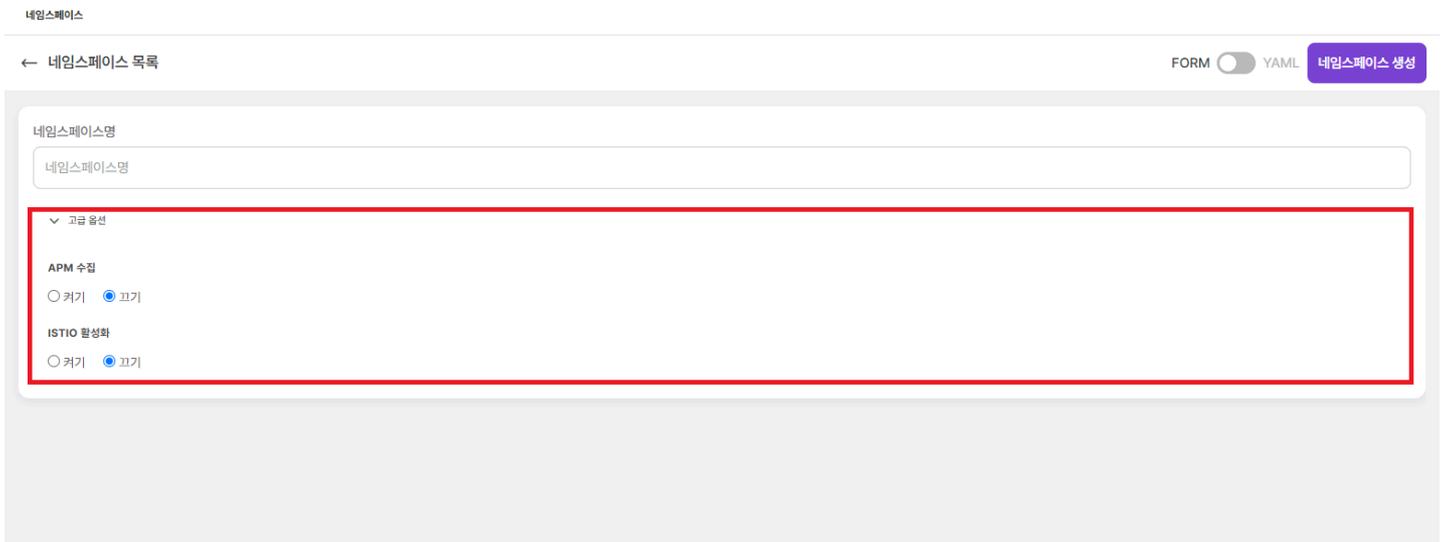


Table 7. YAML

항목	설명
수정	편집기에서 수정한 내용을 반영

4.2.2.1. 네임스페이스 생성

우측 상단의 + 네임스페이스 생성 버튼을 선택하여 네임스페이스 생성에 필요한 정보를 입력할 수 있다.



네임스페이스 이름만으로도 생성할 수 있지만 고급 옵션을 이용하면 APM 관련 설정이 가능하다.

항목	설명
APM 수집	네임스페이스 내 스카우터 에이전트의 타겟 수집 서버 설정 (타겟 수집 서버의 네임스페이스를 선택)
ISTIO 활성화	istio-injection 활성화/비활성

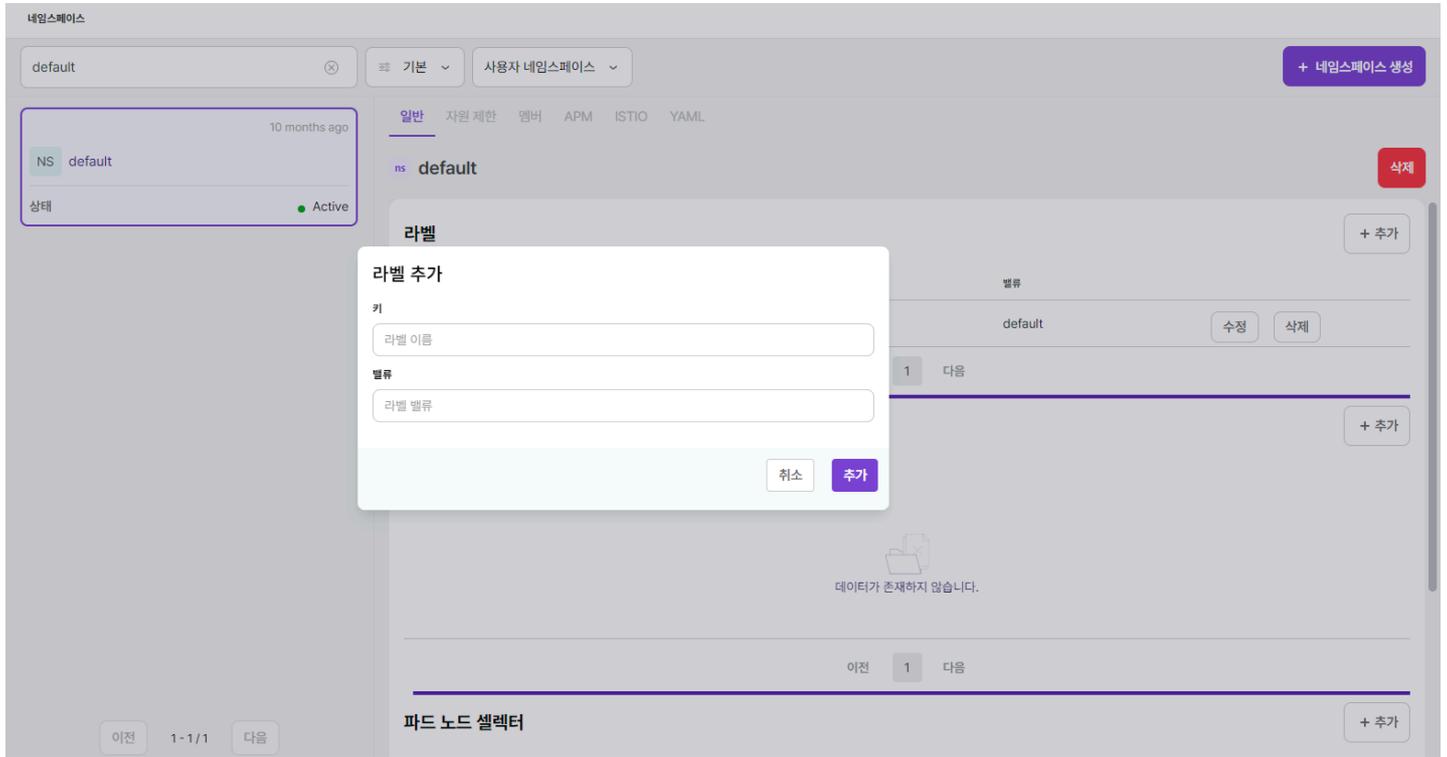
NOTE

스카우터 에이전트를 사용하는 Java 애플리케이션(Tomcat, Wildfly, Springboot 등)의 경우,
Java 1.7 이하 버전은 지원하지 않는다.

4.2.2.2. 네임스페이스 수정

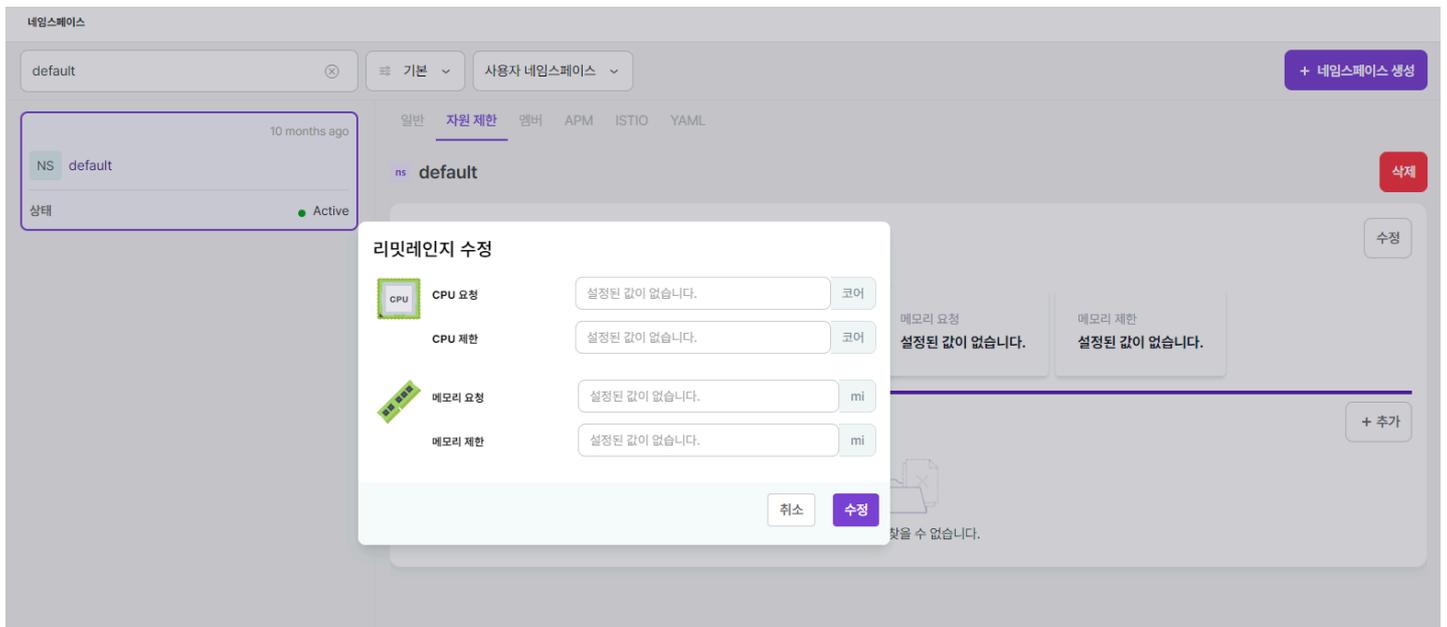
일반 정보 설정

네임스페이스 레이블, 어노테이션, 파드노드셀렉터를 변경할 수 있다. 항목별로 + 추가, 수정, 삭제 버튼을 이용해 값을 설정하거나, 수정/삭제 한다. 각 항목에 등록 시에는 키/값 형태로 설정한다.



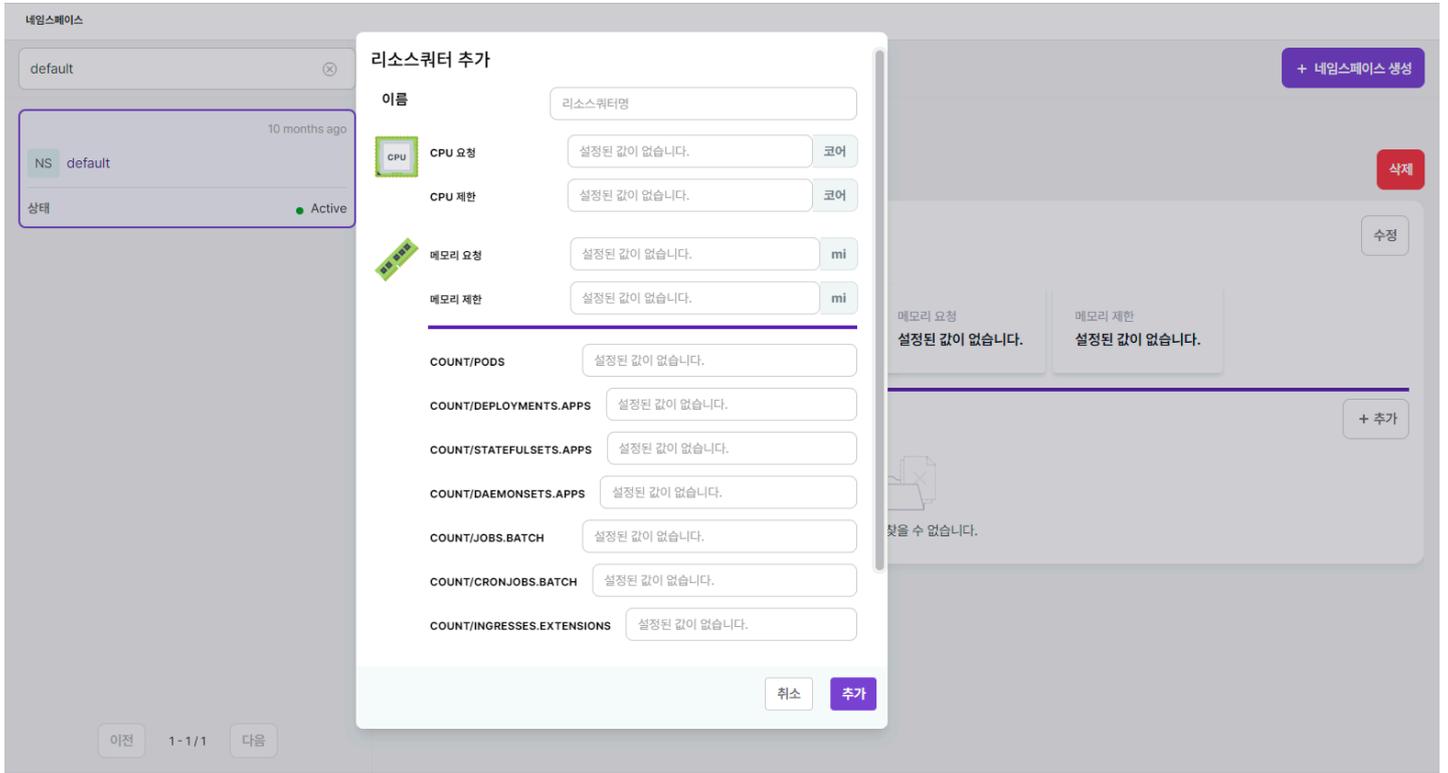
자원 제한 설정

쿠버네티스의 리밋레인지와 리소스쿼터를 이용해 네임스페이스 별로 자원을 제한할 수 있다.



항목	설명
CPU 요청	컨테이너별 최소 CPU 요구량
CPU 제한	컨테이너별 최대 CPU 사용량

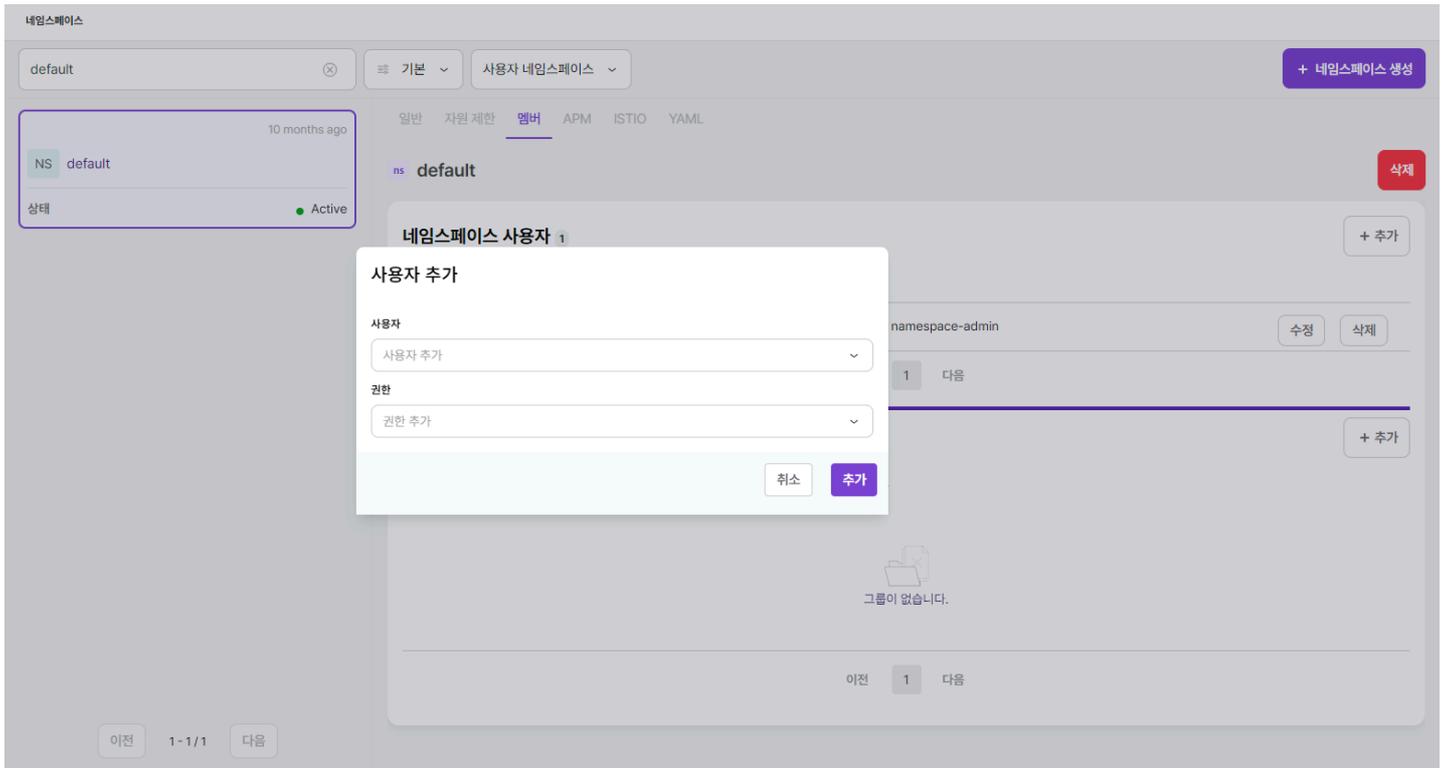
항목	설명
메모리 요청	컨테이너별 최소 메모리 요구량
메모리 제한	컨테이너별 최대 메모리 사용량



항목	설명
이름	리소스쿼터 이름
CPU 요청	네임스페이스에 배포하는 파드들에 대해 CPU 요구량의 총량을 제한
CPU 제한	네임스페이스에 배포하는 파드들에 대해 CPU 최대 사용량의 총량을 제한
메모리 요청	네임스페이스에 배포하는 파드들에 대해 메모리 요구량의 총량을 제한
메모리 제한	네임스페이스에 배포하는 파드들에 대해 메모리 최대 사용량의 총량을 제한
COUNT/PODS	네임스페이스에 배포할 수 있는 파드의 총 수
COUNT/DEPLOYMENTS.APPS	네임스페이스에 배포할 수 있는 디플로이먼트의 총 수
COUNT/STATEFULSETS.APPS	네임스페이스에 배포할 수 있는 스테이트풀셋의 총 수
COUNT/DAEMONSETS.APPS	네임스페이스에 배포할 수 있는 데몬셋의 총 수
COUNT/JOBS.BATCH	네임스페이스에 배포할 수 있는 잡의 총 수
COUNT/CRONJOBS.BATCH	네임스페이스에 배포할 수 있는 크론잡의 총 수
COUNT/INGRESSES.EXTENSIONS	네임스페이스에 배포할 수 있는 인그레스의 총 수
COUNT/SERVICES	네임스페이스에 배포할 수 있는 서비스의 총 수
COUNT/PERSISTENTVOLUMECLAIMS	네임스페이스에 배포할 수 있는 퍼시스턴트볼륨클레임의 총 수

멤버 설정

네임스페이스를 사용할 수 있는 사용자와 그룹을 설정한다. 사용자/그룹 별로 각각 권한을 설정할 수 있다. + 추가 버튼을 선택하여 사용자/그룹을 추가하거나 개별 사용자/그룹의 우측에 있는 수정, 삭제 버튼을 이용해 변경할 수 있다.

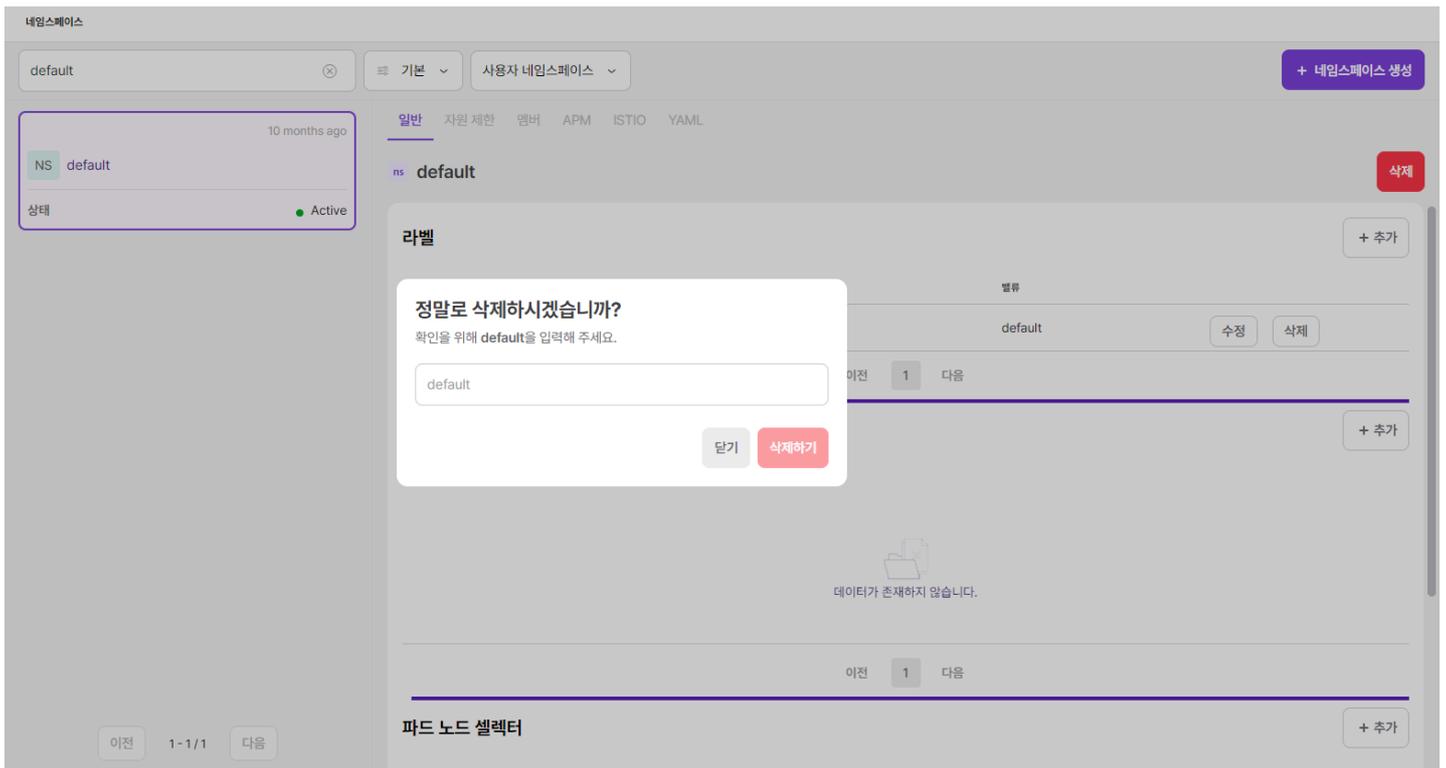


APM 설정

IMPORTANT | APM 설정은 네임스페이스 배포 시 설정한 값을 변경할 수 없다.

4.2.2.3. 네임스페이스 삭제

우측 상단의 삭제 버튼을 선택하여 해당 네임스페이스를 삭제할 수 있다.



4.2.3. 노드

클러스터는 한개 이상의 노드로 구성되며 노드는 물리적 또는 가상 머신을 의미한다. 쿠버네티스는 파드를 노드에서 실행한다. 각 노드는 컨트롤 플레인에 의해 관리되며 파드 실행에 필요한 서비스를 포함한다. 노드는 역할에 따라 마스터(또는 컨트롤 플레인) 노드와 워커 노드로 구분된다.

노드

이름 ↑	상태	롤	파드	CPU	메모리	
dev-accordion1	● Ready	control-plane	14% 15/110	4% 0.324/8 Core	40% 3.083/7.795 Gi	Cordon Drain 라벨 수정
dev-accordion2	● Ready	control-plane	11% 12/110	3% 0.257/8 Core	49% 3.801/7.795 Gi	Cordon Drain 라벨 수정
dev-accordion3	● Ready	control-plane	12% 13/110	5% 0.373/8 Core	53% 4.126/7.795 Gi	Cordon Drain 라벨 수정
dev-accordion4	● Ready	worker	40% 80/200	4% 0.664/16 Core	55% 25.766/47.168 Gi	Cordon Drain 라벨 수정
dev-accordion5	● Ready	worker	66% 131/200	5% 0.804/16 Core	70% 32.848/47.168 Gi	Cordon Drain 라벨 수정

이전 1 다음

노드 메뉴에서 제공하는 정보는 다음과 같다.

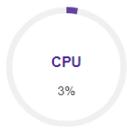
항목	설명
이름	노드 이름
상태	노드 상태 (정상일 경우 Ready 로 표시)
롤	노드 역할 (control-plane 또는 worker)
파드	노드에 배포된 파드 현황 및 파드 상세 목록
CPU	노드에서 사용중인 CPU 현황
메모리	노드에서 사용중인 메모리 현황
노드 이벤트	최근 1시간 이내에 노드에 발생한 이벤트 목록

보다 상세한 정보를 원하는 경우 해당 노드를 선택하면 상세 정보 페이지를 확인할 수 있다.

노드 > 상태

Cordon Drain 라벨 수정

acc-master (10.20.200.221)



CPU
3%



메모리
18%



디스크
40%

CPU
0.26/8.00 Core

메모리
2.83/15.64 Gi

디스크
77.15/191.02 Gi

파드
13/200

총 이미지 수
28

노드

생성된 시간	2024-06-20 17:11:05
IP	10.20.200.221
운영체제	linux
Machine ID	593c32fd97424d9d99b7cdee3978db45
Boot ID	58f21a9d-c5e7-4ec4-a316-138b6d430767
버전	4.18.0-553.el8_10.x86_64
컨테이너 런타임 버전	containerd://1.6.28
Kubelet 버전	v1.29.5
상태	True
원인	KubeletReady
메시지	kubelet is posting ready status
마지막 하트비트	2024-09-06 08:35:57
마지막 상태 변화	2024-07-03 19:50:08

라벨

```
beta.kubernetes.io/arch=amd64
beta.kubernetes.io/os=linux
kubernetes.io/arch=amd64
kubernetes.io/hostname=acc-master
kubernetes.io/os=linux
node-role.kubernetes.io/control-plane=
node.kubernetes.io/exclude-from-external-load-balancers=
```

어노테이션

```
kubeadm.alpha.kubernetes.io/cri-socket=unix:///run/containerd/containerd.sock
kubectrl.kubernetes.io/last-applied-configuration={"apiVersion":"v1","kind":"Node","metadata":{"annotations":{"kubeadm.alpha.kubernetes.io/cri-socket":"unix:///run/containerd/containerd.sock"},"node.alpha.kubernetes.io/ttl":"0"},"projectcalico.org/gIIPv4Address":"10.20.200.221/16","projectcalico.org/IPv4IPIPtunnelAddr":"172.32.183.128","volumes.kubernetes.io/controller-managed-attach-detach":"true"},"creationTimestamp":"2024-06-20T08:11:05Z","labels":{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kubernetes.io/arch":"amd64","kubernetes.io/hostname":"acc-master","kubernetes.io/os":"linux","node-role.kubernetes.io/control-plane":"","node.kubernetes.io/exclude-from-external-load-balancers":""},"name":"acc-master","resourceVersion":"3219853","uid":"932ba777-7b7e-4ccd-aea5-7137c90b0f4b"},"spec":{"podCIDR":"172.32.0.0/24","podCIDRs":["172.32.0.0/24"],"taints":[{"effect":"NoSchedule","key":"node-role.kubernetes.io/control-plane"}],"status":{"addresses":[{"address":"10.20.200.221","type":"InternalIP"}],"address":"acc-master","type":"Hostname"},"allocatable":{"cpu":"8","ephemeral-storage":"184591738984","hugepages-1Gi":"0","hugepages-2Mi":"0","memory":"16298156Ki","pods":"200"},"capacity":{"cpu":"8","ephemeral-storage":"200294856Ki","hugepages-1Gi":"0","hugepages-2Mi":"0","memory":"16400556Ki","pods":"200"},"conditions":[{"lastHeartbeatTime":"2024-07-03T10:31:38Z","lastTransitionTime":"2024-07-03T10:31:38Z","message":"Calico is running on this node","reason":"CalicoSUp","status":"False","type":"NetworkUnavailable"},"lastHeartbeatTime":"2024-07-03T10:47:31Z","lastTransitionTime":"2024-06-20T08:10:57Z","message":"kubel et has sufficient memory available","reason":"KubeletHasSufficientMemory","status":"False","type":"MemoryPressure"},"lastHeartbeatTime":"2024-07-03T10:47:31Z","lastTransitionTime":"2024-06-20T08:10:57Z","message":"kubel et has no disk pressure","reason":"KubeletHasNoDiskPressure","status":"False","type":"DiskPressure"},"lastHeartbeatTime":"2024-07-03T10:47:31Z","lastTransitionTime":"2024-06-20T08:10:57Z","message":"kubel et has sufficient PID available","reason":"KubeletHasSufficientPID","status":"False","type":"PIDPressure"},"lastHeartbeatTime":"2024-07-03T10:47:31Z","lastTransitionTime":"2024-07-03T10:47:31Z","message":"kubel et is posting ready status","reason":"KubeletReady","status":"True","type":"Ready"},"nodeName":"acc-master","nodeSystemInfo":{"bootID":"58f21a9d-c5e7-4ec4-a316-138b6d430767","containerRuntimeVersion":"containerd://1.6.28","kernelVersion":"4.18.0-553.el8_10.x86_64","operatingSystem":"linux","osImage":"Red Hat Enterprise Linux 8.10 (Green Obsidian)"}}}
```

노드 이벤트

파드

Age	네임스페이스	이름	상태	Ready	파드IP	재시작
6 hours ago	acc-system	pgdata-backup-node-28759260-1gdivp	Succeeded	0/2	10.20.200.221	2
4 weeks ago	kube-system	kube-proxy-4rd2v	Running	1/1	10.20.200.221	4
1 month ago	kube-system	coredns-97bd8d8c5-qkcsz	Running	1/1	172.32.183.176	5
2 months ago	kube-system	kube-scheduler-acc-master	Running	1/1	10.20.200.221	5
2 months ago	kube-system	kube-controller-manager-acc-master	Running	1/1	10.20.200.221	5
2 months ago	kube-system	kube-apiserver-acc-master	Running	1/1	10.20.200.221	9
2 months ago	acc-system	filebeat-filebeat-g2q2v	Running	1/1	172.32.183.174	5
2 months ago	acc-system	node-exporter-7bt28	Running	2/2	10.20.200.221	5
2 months ago	kube-system	calico-kube-controllers-665d779c9-8khp	Running	1/1	172.32.183.175	7
2 months ago	kube-system	etcd-acc-master	Running	1/1	10.20.200.221	6
2 months ago	kube-system	calico-node-bzvv	Running	1/1	10.20.200.221	5
2 months ago	acc-global	gateway-proxy-s9g9s	Running	1/1	172.32.183.173	10
2 months ago	acc-system	accordion-data-provisioner-5d4789cddb-sc4s9	Running	1/1	172.32.183.172	8

노드 메뉴(또는 노드 상세 화면)에서는 각 노드에 대해 Cordon, Drain 명령이 가능하고 노드의 레이블을 변경할 수 있다.

TIP

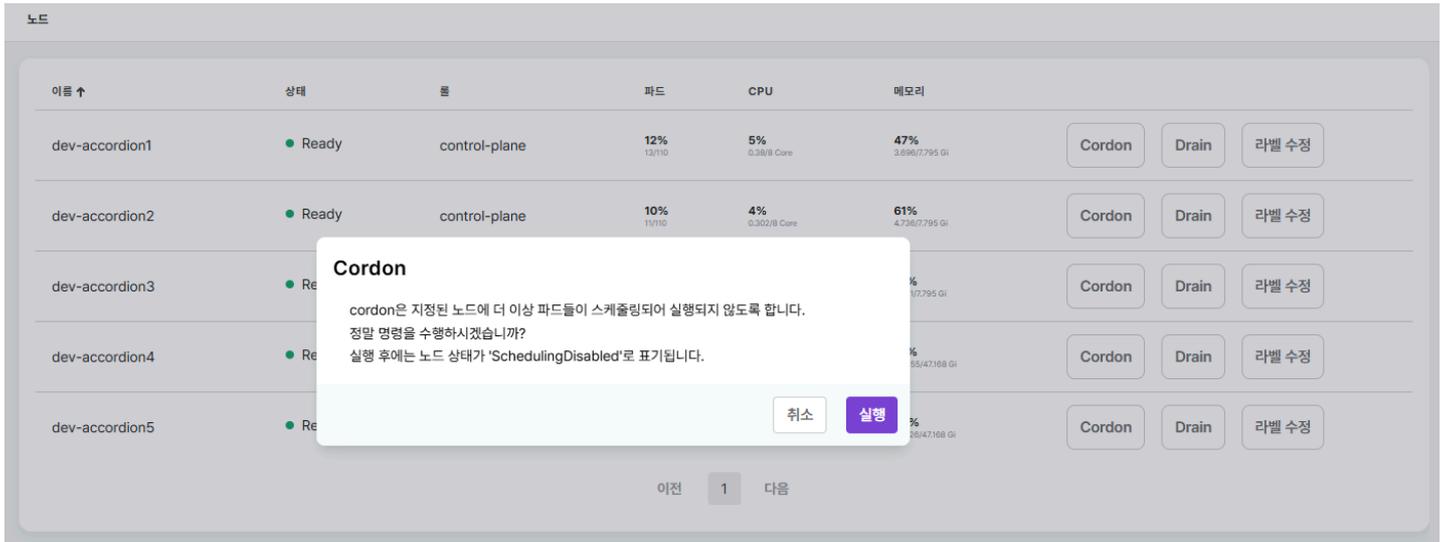
문서에서는 노드 메뉴에서 설정하는 방법을 기준으로 가이드한다.

https://10.60.200.221:30000/docs/

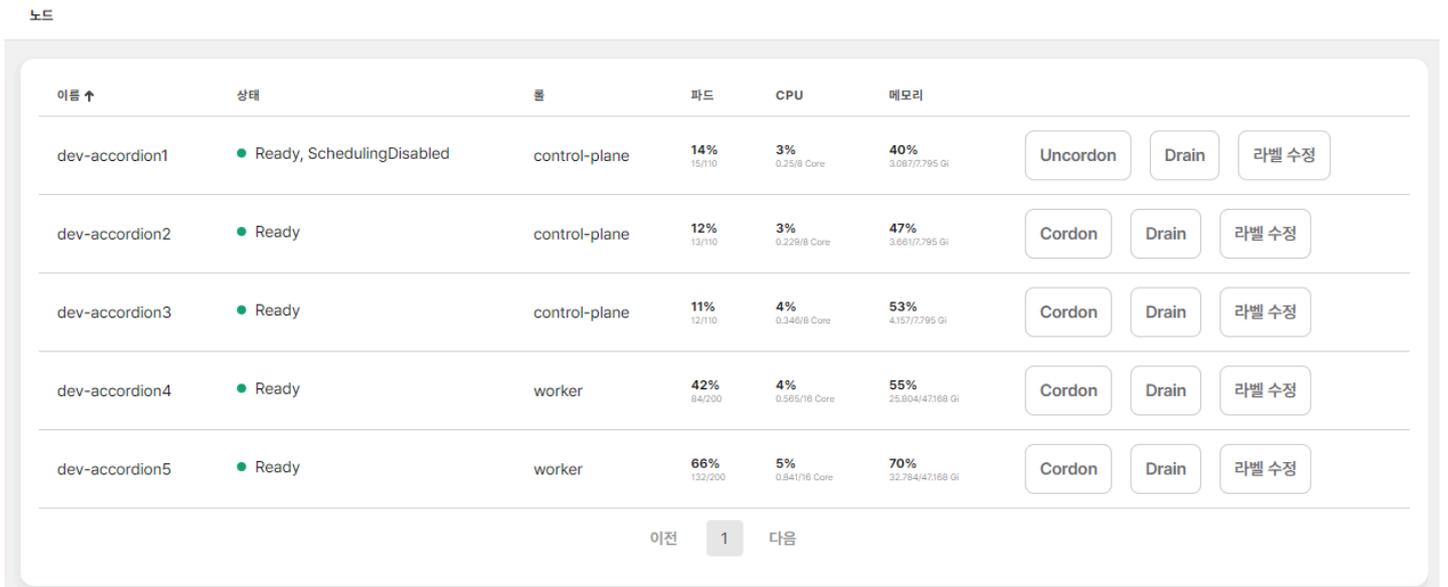
66/349

4.2.3.1. Cordon

Cordon 버튼을 이용해 해당 노드를 쿠버네티스 리소스 배포 대상에서 제외하여 파드가 배포되지 않도록 설정할 수 있다. 이 경우 노드 상태에 SchedulingDisabled 가 추가된다.



Cordon 으로 쿠버네티스 리소스 배포 대상에서 제외된 노드는 Uncordon 버튼을 이용해 다시 쿠버네티스 리소스 배포 대상으로 추가할 수 있다. 쿠버네티스 리소스 배포 대상으로 추가되면 SchedulingDisabled 롤이 제거된다.

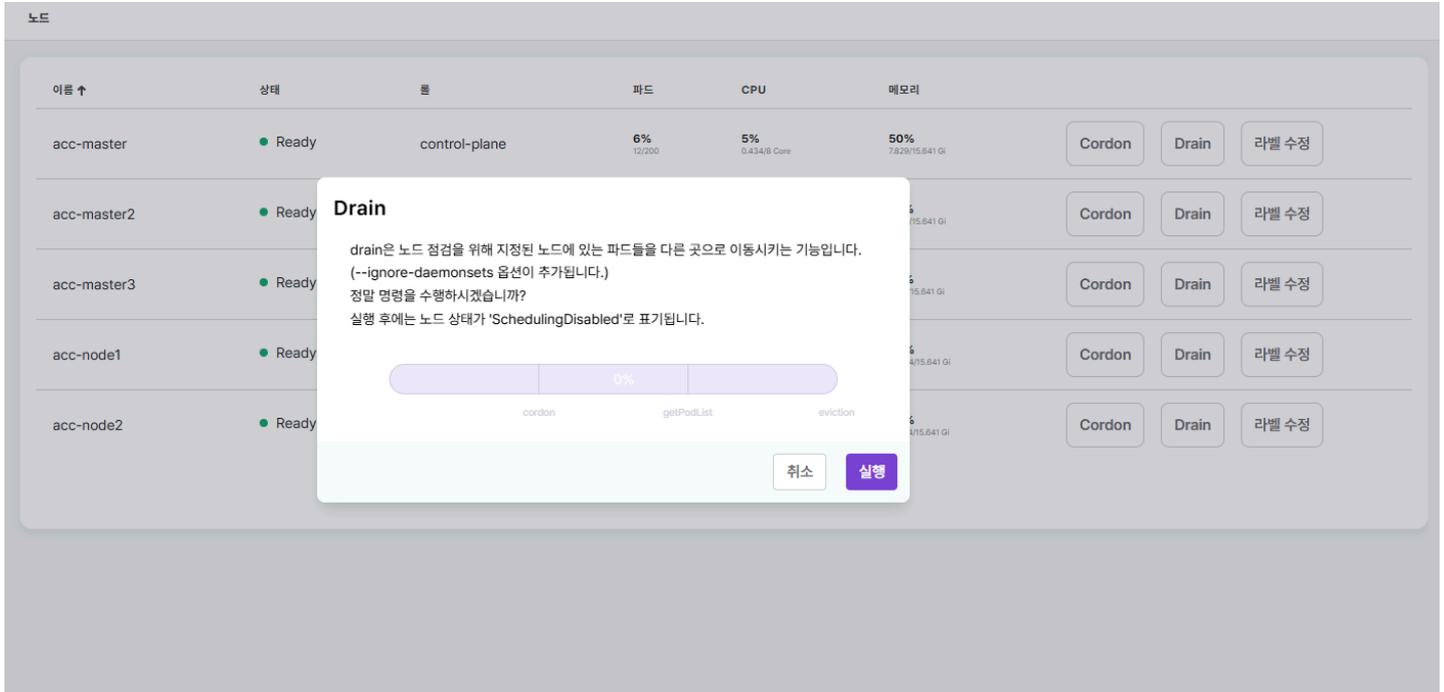


4.2.3.2. Drain

Drain 버튼을 이용해 노드에 배포된 파드를 다른 노드로 이동시킬 수 있다. 이 경우 Cordon 과 마찬가지로 노드의 상태에 SchedulingDisabled 가 추가된다.

IMPORTANT

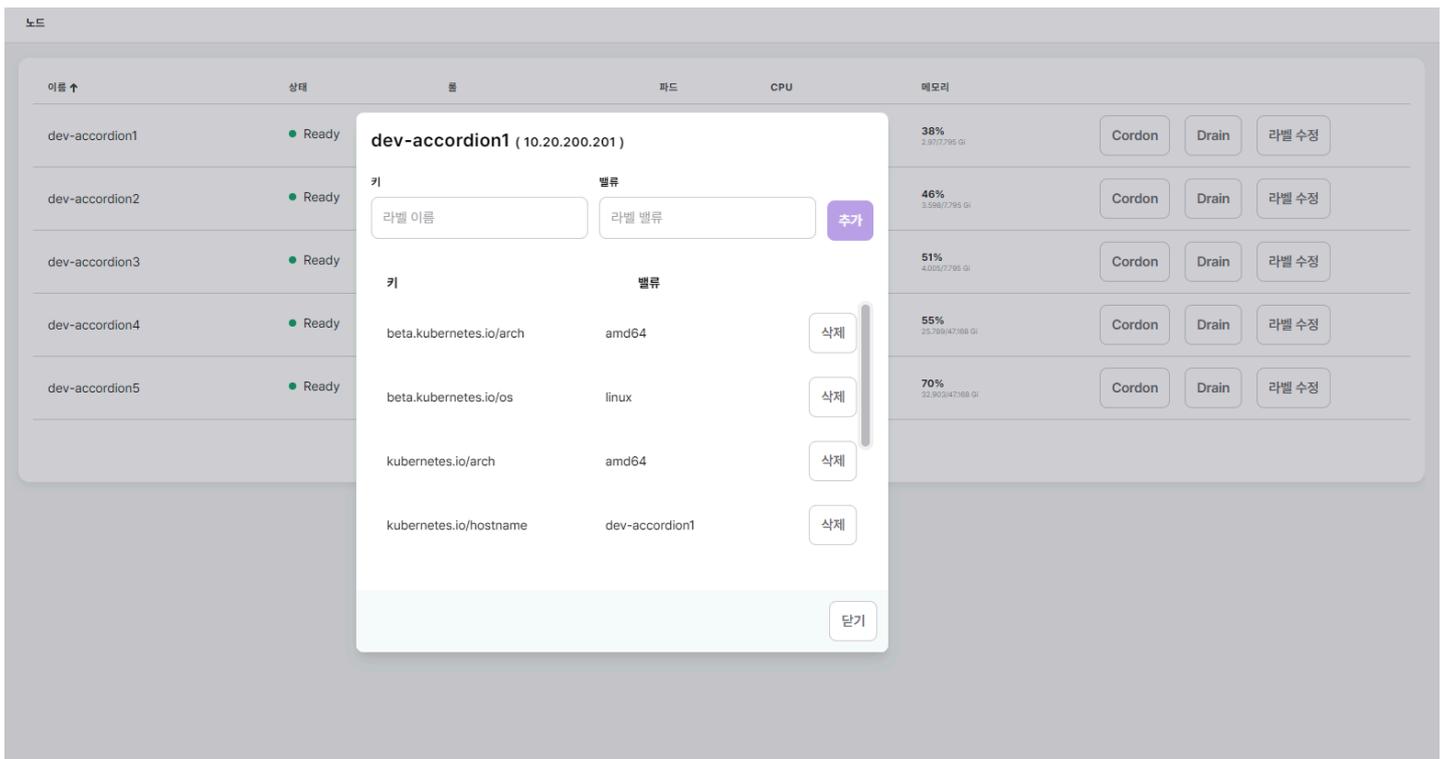
데몬셋으로 배포된 파드의 경우는 예외로 다른 파드로 이동하지 않는다.



Drain 으로 파드가 다른 노드로 이동하는 동안 Drain 버튼을 선택하면 진행 상황을 확인할 수 있다.

4.2.3.3. 라벨 수정

노드의 레이블을 변경한다.



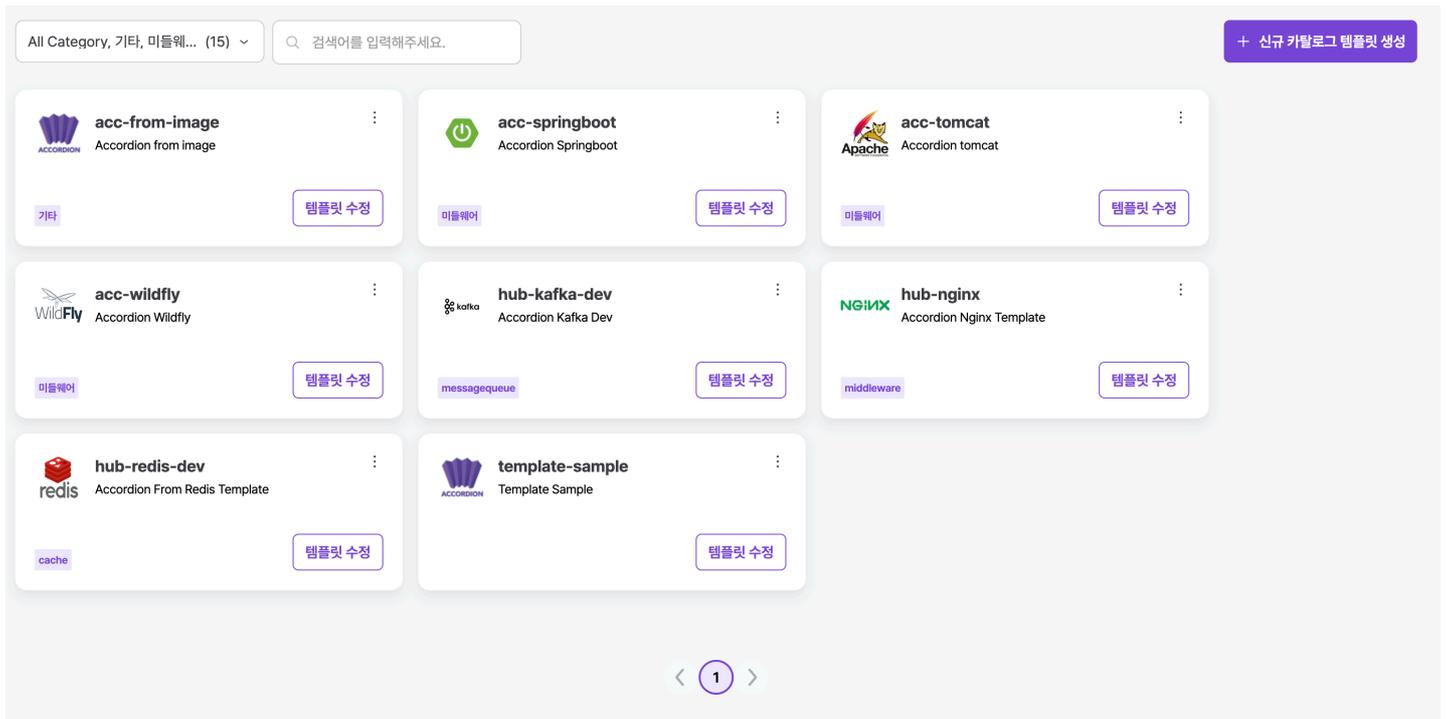
4.2.4. 애플리케이션

4.2.4.1. 클러스터 카탈로그 템플릿

사용자는 애플리케이션 메뉴의 클러스터 카탈로그 템플릿을 이용해 카탈로그를 위한 클러스터 레벨의 템플릿을 작성할 수 있다. 클러스터 레벨에서 공통으로 사용할 수 있는 카탈로그 템플릿으로 카탈로그를 생성할 때 사용된다.

TIP

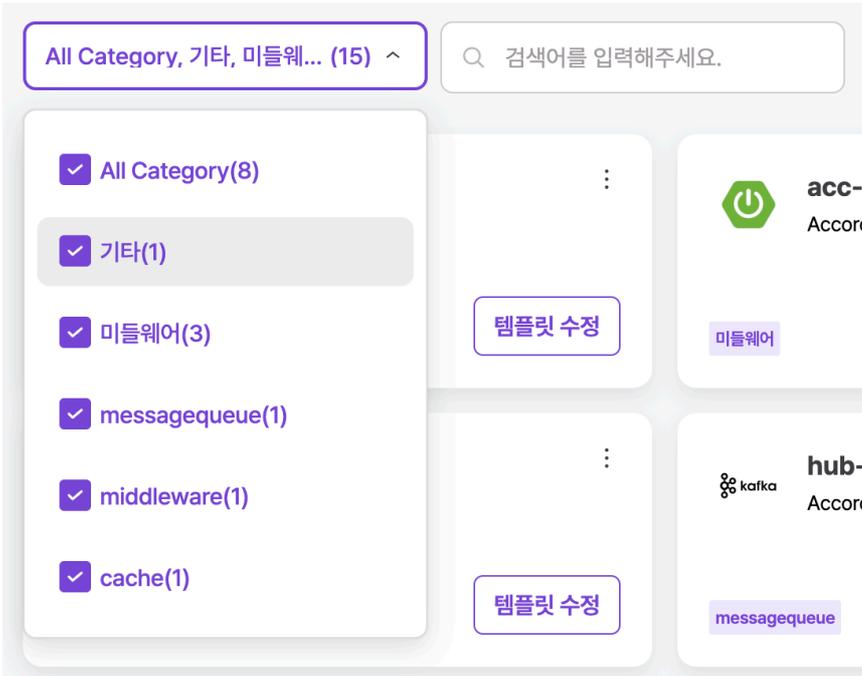
- 네임스페이스 스코프의 카탈로그 메뉴에서 카탈로그를 생성할 때 카탈로그 템플릿을 직접 생성할 수도 있다.
- 미리 카탈로그 템플릿을 만들어 두면 카탈로그를 좀 더 간단하게 생성할 수 있다.



표시되는 카드에 나타나는 정보는 다음과 같다.

항목	설명
이미지	클러스터 카탈로그 템플릿 생성 시 등록된 이미지 표시
이름	해당 클러스터 카탈로그 템플릿의 이름으로 가운데 상단에 위치
요약	해당 클러스터 카탈로그 템플릿의 요약 설명으로 중앙에 위치
카테고리	카탈로그 YAML의 annotations 중 <code>ui.accordions.co.kr/category: <카테고리></code> 형식으로 입력된 텍스트가 표시된다.
스코프	네임스페이스 레벨인 경우 네임스페이스 라벨이 하단에 위치

템플릿 카테고리 필터 및 검색

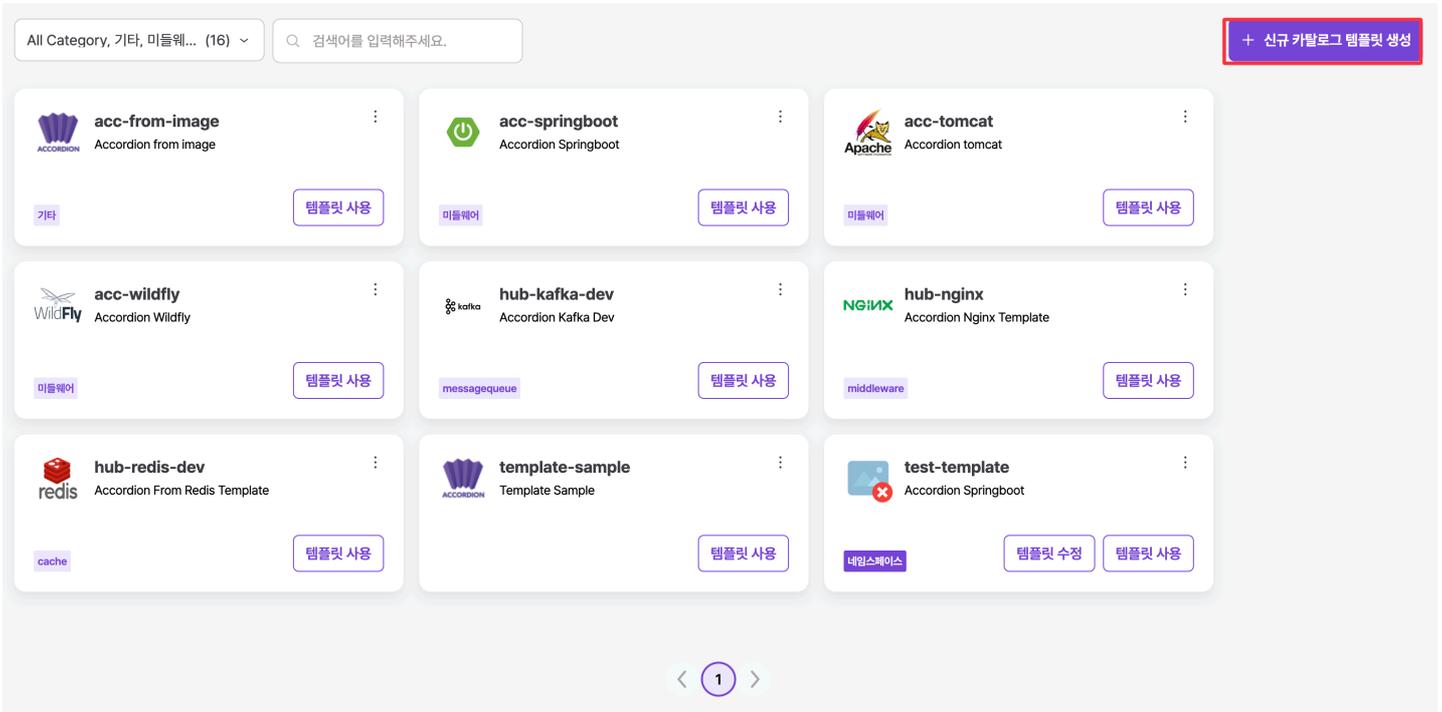


카탈로그 템플릿 카테고리 필터 및 검색 기능을 제공한다.

TIP

카테고리는 카탈로그 YAML의 annotations 중 `ui.accordions.co.kr/category: <카테고리>` 형식으로 입력된 텍스트를 기준으로 한다.

템플릿 생성



신규 카탈로그 템플릿 생성 이라는 이름을 가진 첫 번째에 위치하고 있는 카드의 템플릿 생성 버튼을 선택하여 새로운 템플릿을 생성할 수 있다.

일반 설정

애플리케이션 > 클러스터 카탈로그 템플릿 > 목록

← 일반 설정 파이프라인 설정 배포 리소스 설정 YAML 생성

이름

요약

상세 설명
 Write Preview

로고 이미지

 이미지 업로드 (PNG 형식만 지원)

입력하는 카탈로그 템플릿 기본 정보는 다음과 같다.

항목	설명
이름	클러스터 카탈로그 템플릿의 이름
요약	템플릿에 대한 간단한 설명(생성 후 목록 정보에 표시)
상세 설명	템플릿에 대한 자세한 설명(생성 후 상세 설명 버튼 선택시 표시)
로고 이미지	템플릿에 대한 로고 이미지(미설정 시 템플릿의 기본 로고로 설정)

파이프라인 설정

← 일반 설정 파이프라인 설정 배포 리소스 설정 YAML 생성

파이프라인 템플릿 선택

파이프라인 미리보기


템플릿에 기본적으로 들어갈 파이프라인 템플릿을 설정한다. 파이프라인 템플릿은 빌드 탭의 클러스터 파이프라인 템플릿에서 생성한다.

모든 설정이 끝났으면 마지막으로 우측 상단의 생성 버튼을 선택하여 템플릿을 생성한다.

배포 리소스 설정

애플리케이션 > 클러스터 카탈로그 템플릿 > 목록

← 일반 설정 파이프라인 설정 **배포 리소스 설정** YAML

생성

RESOURCE: RESOURCE VALUES

```

1 cm:
2   data:
3     - key: os
4       value: linux
5     - key: arch
6       value: amd64
7

```

RESOURCE: RESOURCE YAML

```

1 apiVersion: v1
2 kind: ConfigMap
3 metadata:
4   name: {{{.CATALOG_NAME}}}
5 data:
6   {{- range $i, $e := .values.cm.data}}
7     {{{$.key}}}: {{{$.value}}}
8   {{- end}}

```

RESOURCE: VALUESHEMA YAML

```

1 type: object
2 properties:
3   cm:
4     type: object
5     properties:
6       data:
7         type: array
8         description: 데이터
9         items:
10          type: object
11          properties:
12            key:
13              type: string
14              pattern: ^[-_a-zA-Z0-9]+$
15              maxLength: 253
16            value:
17              type: string
18              format: textarea
19

```

배포 리소스 설정은 yaml 에디터로 작성한다. 작성해야할 항목은 세 가지이다.

항목	설명
values.yaml	템플릿으로 전달되는 값을 정의한다. 해당 값은 resource.yaml 에서 사용되어진다.
resource.yaml	쿠버네티스에 배포할 리소스들에 대한 스펙을 yaml로 정의한다. 여기에서는 카탈로그에서 사용되는 환경 변수와 valueschema를 사용하여 정의할 수 있다.
valueschema.yaml	resource.yaml에서 사용할 valueschema를 정의한다.

TIP

- valueschema의 사용법은 빌드 탭의 클러스터 태스크 템플릿의 valueschema를 참고한다.

YAML

위 설정 탭을 이용하지 않고, YAML 탭에서 템플릿 전체를 수정할 수 있다. 위 경우는 다른 템플릿 내용을 복사/붙여넣기 하기에 용이하다.

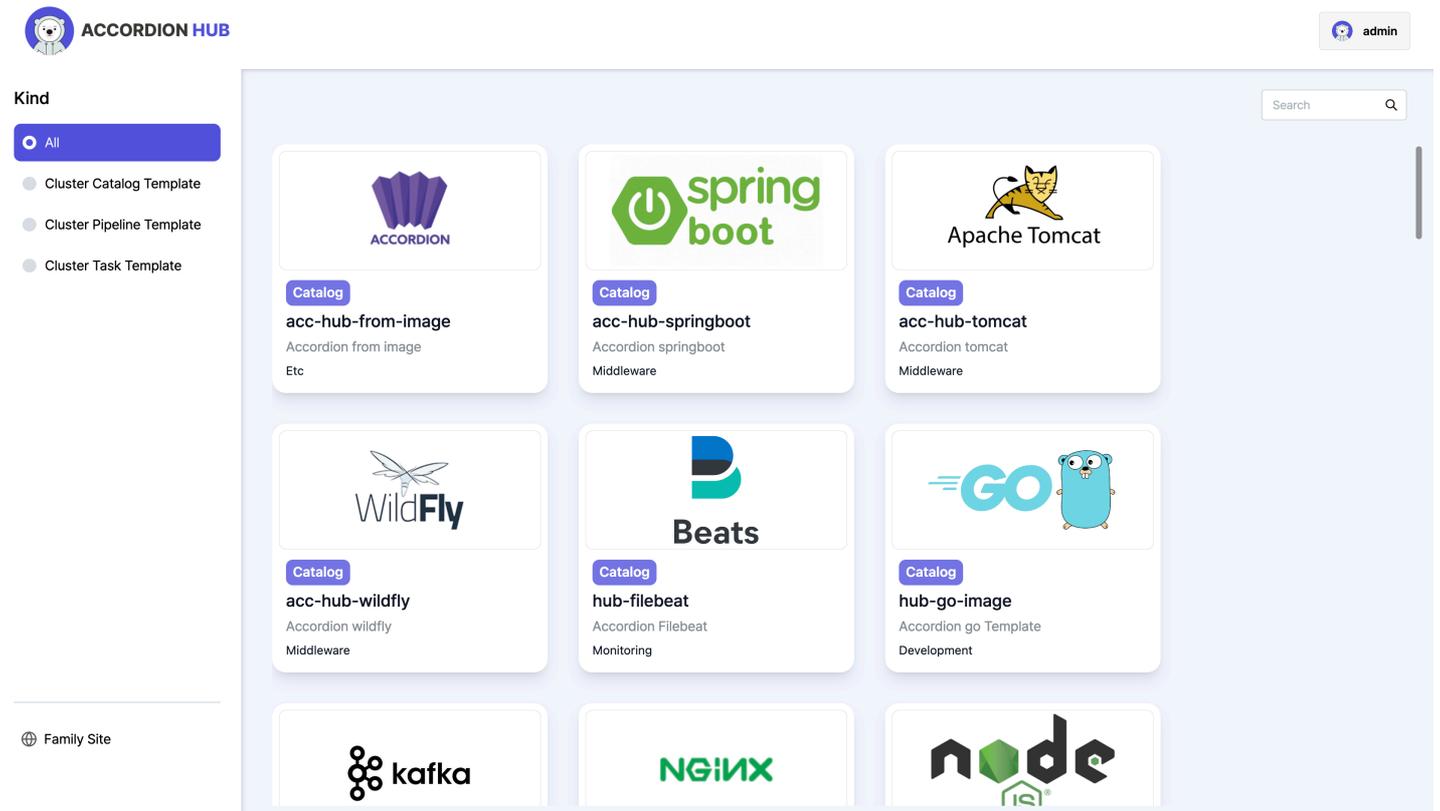
```

1  apiVersion: ckd.accordions.co.kr/v1beta1
2  kind: ClusterCatalogTemplate
3  metadata:
4    annotations:
5      accordions.co.kr/summary: ''
6      accordions.co.kr/description: ''
7    name: ''
8    labels: {}
9  spec:
10 pipelineTemplateRef:
11   name: acc-vcs-jbuild-tomcat
12   clusterScope: true
13   deployStrategy:
14     clusters: []
15     defaultPolicy: Apply
16   image:
17     archiveCount: 5
18     registryName: user-registry
19   resourceValues:
20     - name: resource
21       values:
22         cm:
23           data:
24             - key: os
25               value: linux
26             - key: arch
27               value: amd64
28   template:
29     resources:
30       - name: resource
31         spec: |-
32           apiVersion: v1
33           kind: ConfigMap
34           metadata:
35             name: {{{.CATALOG.NAME}}}
36           data:
37             {{- range $i, $e := .values.cm.data}}
38               {{$e.key}}: {{$e.value}}
39             {{- end}}
40           valueschema:
41             type: object
42             properties:
43               cm:
44                 type: object
45                 properties:
46                   data:

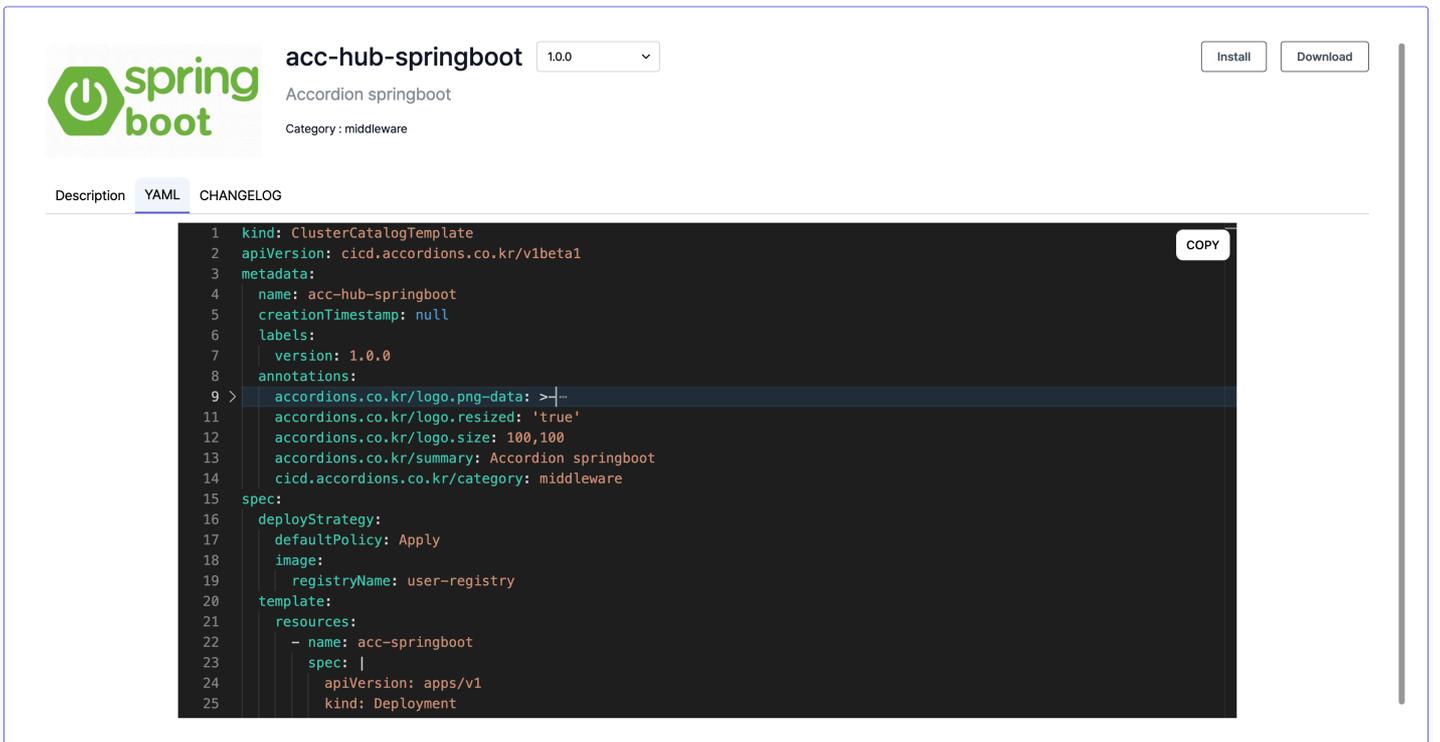
```

다양한 템플릿들을 제공하는 [아코디언 허브 사이트](https://hub.accordions.co.kr/) (<https://hub.accordions.co.kr/>) 를 운영 중으로 사이트 내의 템플릿 YAML을 복사/붙여넣기 하여 템플릿을 손쉽게 생성할 수 있다.

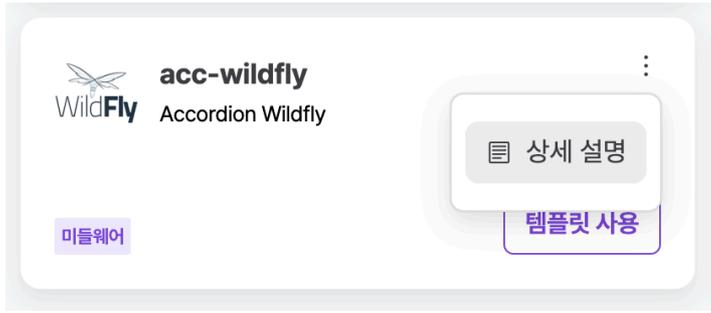
메인화면



상세화면



상세 설명



조회할 클러스터 카탈로그 템플릿 카드에 위치한 상세 설명 버튼을 선택하여 해당 클러스터 카탈로그 템플릿에 대한 상세 설명을 조회할 수 있다.

클러스터 host-cluster-220 네임스페이스 twkang

애플리케이션 > 카탈로그 > 생성

All Category, 기타, 미들웨어... (16)

acc-from-image
Accordion from image

기타 템플릿 사용

미들웨어

acc-springboot
Accordion Springboot

미들웨어

acc-wildfly
Accordion Wildfly

미들웨어 템플릿 사용

hub-kafka-dev
Accordion Kafka Dev

messagequeue

hub-redis-dev
Accordion From Redis Template

cache 템플릿 사용

template-sample
Template Sample

> 이벤트 (NORMAL 250 / WARNING 40)

acc-wildfly ×

acc-wildfly

가벼우면서 강력한 모듈화된 웹 애플리케이션 서버인 WildFly를 기본으로 하여 소스빌드(war), 컨테이너 이미지 빌드, 클러스터에 배포 과정(CI/CD)을 포함한 아코디언에서 제공하는 카탈로그입니다.

1. 스펙

- WildFly
- J2EE8 Specificaion
- Support for JDK8, 11
- Session Clustering(jgroups , kubernetes.KUBE_PING)
- 기본 메모리 설정
 - 컨테이너 1.5 GB (resource.limit=1.5Gi)
 - 할당된 컨테이너의 80% 를 힙메모리 사용 (-XX:InitialRAMPercentage=80.0 -XX:MaxRAMPercentage=80.0)

2. 배포리소스

- Deployemnt
- Service

템플릿 수정

애플리케이션 > 클러스터 카탈로그 템플릿 > 목록

All Category, 기타, 미들웨어... (15)

[+ 신규 카탈로그 템플릿 생성](#)

acc-from-image
Accordion from image

기타 템플릿 수정

acc-springboot
Accordion Springboot

미들웨어 템플릿 수정

acc-tomcat
Accordion tomcat

미들웨어 템플릿 수정

acc-wildfly
Accordion Wildfly

미들웨어 템플릿 수정

hub-kafka-dev
Accordion Kafka Dev

messagequeue 템플릿 수정

hub-nginx
Accordion Nginx Template

middleware 템플릿 수정

hub-redis-dev
Accordion From Redis Template

cache 템플릿 수정

template-sample
Template Sample

템플릿 수정

< 1 >

기본 클러스터 카탈로그 템플릿을 수정하기 위해서는 대상 템플릿 카드의 우측 하단에 있는 템플릿 수정 버튼을 선택한다.

애플리케이션 > 클러스터 카탈로그 템플릿 > 목록

← [일반 설정](#) [파이프라인 설정](#) [배포 리소스 설정](#) [YAML](#) 수정

이름

요약

상세 설명
[Write](#) [Preview](#)

```
acc-springboot
----
...

SprintBoot기반으로 개발된 애플리케이션을 지원하기 위해 아코디언에서 제공하는 카탈로그입니다.
Spring 프레임워크 기반의 코드의 소스빌드(jar,war), 컨테이너 이미지 빌드, 클러스터 배포 과정(CI/CD)을 포함합니다.
...

1. 스펙
* Support for JDK8, 11
* 기본 메모리 설정
* 컨테이너 `1GB` (`resource.limit=1Gi`)
* 할당된 컨테이너의 `80%` 를 힙메모리 사용 (`-XX:InitialRAMPercentage=80.0 -XX:MaxRAMPercentage=80.0`)

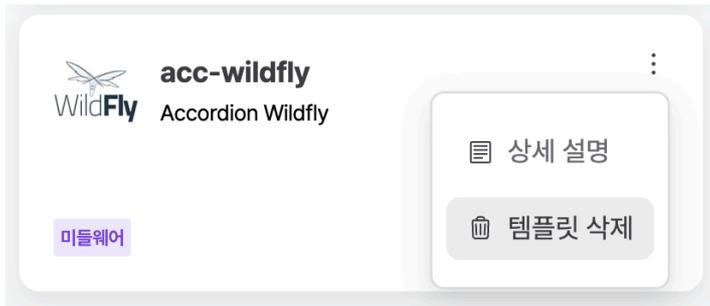
2. 배포리소스
* Deployemnt
* Service
```

로고 이미지

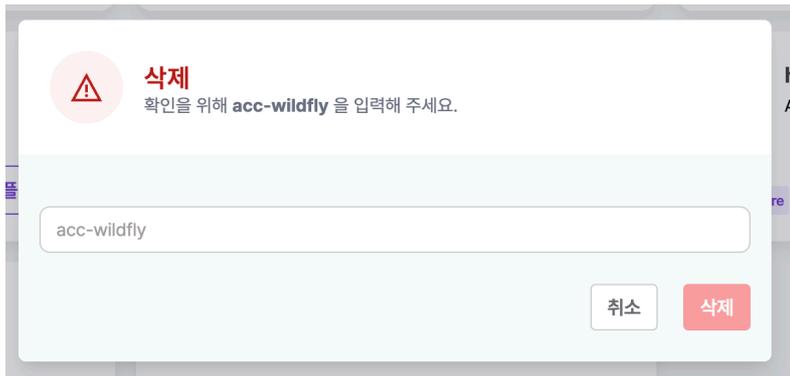
이미지 업로드 (PNG 형식만 지원)

수정하길 원하는 항목을 수정한 다음 우측 상단의 수정 버튼을 선택하여 템플릿 수정을 반영한다.

템플릿 삭제



생성된 클러스터 카탈로그 템플릿을 삭제하기 위해서는 대상 템플릿 카드의 우측 상단의  버튼 클릭 후 템플릿 삭제 버튼을 선택한다.



모달에서 삭제하려는 템플릿의 이름을 입력하여 삭제한다.

4.2.4.2. 헬름

사용자는 애플리케이션 메뉴의 헬름을 이용해 클러스터에 애플리케이션을 배포할 수 있다. 주로 mysql, redis 와 같이 서비스 형태의 애플리케이션을 배포할 때 사용한다.

TIP

- 헬름으로 배포하기 위해서는 글로벌 스코프의 헬름 메뉴에서 리포지터리와 차트를 설정해야 한다.
- 빌드가 필요한 애플리케이션은 네임스페이스 스코프의 카탈로그 메뉴를 이용해 배포한다.

애플리케이션 > 헬름 > 헬름 앱

이름 ↓	네임 스페이스	개정	업데이트	상태	차트	앱 버전	런치
test-aggr	huu	1	2024-03-06 14:33:55	deployed	nginx-15.1.2	1.25.1	삭제
test	default	1	2024-03-26 15:04:22	deployed	nginx-15.1.2	1.25.1	삭제
test	huu	5	2024-03-05 10:13:35	deployed	nginx-15.4.0	1.25.3	삭제
spin-containerd-shim-installer	hwpark	1	2024-02-20 16:24:30	deployed	spin-containerd-shim-installer-0.10.0	0.10.0	삭제
opensearch-dashboards	alert	2	2023-12-27 11:31:38	deployed	opensearch-dashboards-1.3.0	1.3.0	삭제
opensearch	alert	1	2023-12-27 13:05:55	deployed	opensearch-1.9.0	1.3.0	삭제
mysql	mjdev	1	2024-01-04 10:04:16	deployed	mysql-9.14.1	8.0.35	삭제
mysql	alert	1	2023-11-09 14:22:06	deployed	mysql-9.14.1	8.0.35	삭제
locust	ingresstest	5	2024-07-02 22:23:59	deployed	locust-0.31.5	2.15.1	삭제
locust	stress	1	2024-06-27 14:37:24	deployed	locust-0.31.5	2.15.1	삭제

이전 1 2 다음

제공하는 정보는 다음과 같다.

항목	설명
이름	헬름 차트를 이용해 배포한 애플리케이션 이름
네임스페이스	애플리케이션을 배포한 네임스페이스
개정	업데이트 횟수
업데이트	업데이트 시간
상태	배포 상태
차트	애플리케이션 배포 시 사용한 헬름 차트
앱 버전	헬름 차트에 설정된 앱 버전

애플리케이션 배포

헬름으로 애플리케이션을 배포하기 위해서는 우측 상단의 런치 버튼을 선택하면 차트 목록을 조회할 수 있다.

애플리케이션 > 헬름 > 헬름 런치

전체

aqua-quickstart

리포지터리
버전
앱 버전

aquahelm
2022.4
2022.4.1

A Helm chart for Aqua CSP Quick Start

argon-broker

리포지터리
버전
앱 버전

aquahelm
2022.4
1.0.9

A Helm chart for the Argon Broker...

cloud-connector

리포지터리
버전
앱 버전

aquahelm
2022.4
2022.4.5

A Helm chart for Aqua Cloud-Connector

codesec-agent

리포지터리
버전
앱 버전

aquahelm
2022.4
1.2.11

A Helm chart for Aqua supply chain...

cyber-center

리포지터리
버전
앱 버전

aquahelm
2022.4
2022.4.6

A Helm chart for Aqua CyberCenter

enforcer

리포지터리
버전
앱 버전

aquahelm
2022.4
2022.4.23

A Helm chart for the Aqua Enforcer

gateway

리포지터리
버전
앱 버전

aquahelm
2022.4
2022.4.14

A Helm chart for the Aqua Gateway

harbor-scanner-aqua

리포지터리
버전
앱 버전

aquahelm
0.14.0
0.14.0

Harbor scanner adapter for Aqua...

이전
1
2
3
4
5
...
8
다음

배포할 차트를 선택하면 차트의 상세 정보를 조회할 수 있다.

애플리케이션 > 헬름 > 헬름 런치

Bitnami package for Apache Tomcat

Apache Tomcat is an open-source web server designed to host and run Java-based web applications. It is a lightweight server with a good performance for applications running in production environments.

[Overview of Apache Tomcat](#)

Trademarks: This software listing is packaged by Bitnami. The respective trademarks mentioned in the offering are owned by the respective companies, and use of them does not imply any affiliation or endorsement.

TL;DR

```
helm install my-release oci://registry-1.docker.io/bitnamicharts/tomcat
```

Looking to use Apache Tomcat in production? Try [VMware Tanzu Application Catalog](#), the commercial edition of the Bitnami catalog.

Introduction

This chart bootstraps a [Tomcat](#) deployment on a [Kubernetes](#) cluster using the [Helm](#) package manager.

Tomcat implements several Java EE specifications including Java Servlet, JavaServer Pages, Java EL, and WebSocket, and provides a "pure Java" HTTP web server environment for Java code to run in.

Bitnami charts can be used with [Kubeapps](#) for deployment and management of Helm Charts in clusters.

Prerequisites

- Kubernetes 1.23+
- Helm 3.8.0+

tomcat

리포지터리
버전
앱 버전

bitnami
10.1.28
11.2.18

배포하기

배포하기 버튼을 선택하면 배포에 필요한 값을 설정할 수 있다.

YAML 미리보기

```

1 # Copyright Broadcom, Inc. All Rights Reserved.
2 # SPDX-License-Identifier: APACHE-2.0
3
4 ## @section Global parameters
5 ## Global Docker image parameters
6 ## Please, note that this will override the image parameters, including dependencies, configured to use the global value
7 ## Current available global Docker image parameters: imageRegistry, imagePullSecrets and storageClass
8 ##
9
10 ## @param global.imageRegistry Global Docker image registry
11 ## @param global.imagePullSecrets Global Docker registry secret names as an array
12 ## @param global.defaultStorageClass Global default StorageClass for Persistent Volume(s)
13 ## @param global.storageClass DEPRECATED: use global.defaultStorageClass instead
14 ##
15 global:
16   imageRegistry: ""
17   ## E.g.
18   ## imagePullSecrets:
19   ##   - myRegistryKeySecretName
20   ##
21   imagePullSecrets: []
22   defaultStorageClass: ""
23   storageClass: ""
24   ## Compatibility adaptations for Kubernetes platforms
25   ##
26   compatibility:
27     ## Compatibility adaptations for Openshift
28     ##
29     openshift:
30       ## @param global.compatibility.openshift.adaptSecurityContext Adapt the securityContext sections of the deployment to make them comp
31       ##
32       adaptSecurityContext: disabled
33 ## @section Common parameters
34 ##
35

```

tomcat



리포지터리	bitnami
버전	10.1.28
앱 버전	11.2.18

Target Namespace

애플리케이션명

좌측의 YAML 편집기에서 차트를 애플리케이션으로 배포할 때 적용할 값을 설정할 수 있다. 우측의 Target Namespace 와 애플리케이션명 은 배포할 네임스페이스와 배포하는 애플리케이션의 이름을 설정한다. 설정이 완료되면 좌측 상단의 YAML 미리보기 버튼을 눌러 배포될 쿠버네티스 리소스를 미리 확인할 수 있다.

TIP

네임스페이스 스코프의 헬름 메뉴에서 애플리케이션을 배포하는 경우 네임스페이스 선택없이 애플리케이션 이름만 설정한다.

작성을 완료하여 런치 버튼을 선택하면 실제 클러스터에 애플리케이션이 배포된다.

애플리케이션 정보 조회

애플리케이션의 상세 정보를 조회하기 위해서는 애플리케이션 목록에서 대상 애플리케이션을 선택한다. 애플리케이션 상세 화면에서는 리소스, 상태, 히스토리에 대한 정보를 확인할 수 있다.

애플리케이션 > 열람 > mysql

mysql

Resources Status History

스태이트풀셋

# ↑	클러스터	내임 스페이스	이름
1	host-cluster-200	alert	mysql

서비스

# ↑	클러스터	내임 스페이스	이름	타입	클러스터 IP	포트/노드포트	EXTERNAL_IP
1	host-cluster-200	alert	mysql-headless	ClusterIP	None	3306	-
2	host-cluster-200	alert	mysql	ClusterIP	-	3306/TCP	-

컨피그맵

# ↑	클러스터	내임 스페이스	이름
1	host-cluster-200	alert	mysql

시크릿

# ↑	클러스터	내임 스페이스	이름
1	host-cluster-200	alert	mysql

리소스 탭에서는 애플리케이션을 구성하는 쿠버네티스 리소스들에 대한 정보를 제공한다.

TIP

리소스 정보는 애플리케이션 구동에 필수적인 파드와 서비스 등의 정보에 대해서만 조회가 가능하다.

애플리케이션 > 열람 > mysql

mysql

Resources Status History

차트들(를) 선택하세요. 업그레이드

별류

```

1 architecture: standalone
2 auth:
3   createDatabase: true
4   customPasswordFiles: {}
5   database: petstore
6   defaultAuthenticationPlugin: ""
7   existingSecret: ""
8   password: admin
9   replicationPassword: ""
10  replicationUser: replicator
11  rootPassword: root
12  usePasswordFiles: false
13  username: admin
14 clusterDomain: cluster.local
15 commonAnnotations: {}
16 commonLabels: {}
17 diagnosticMode:
18   args:
19     - infinity
20   command:
21     - sleep
22   enabled: false
23 extraDeploy: []
24 fullnameOverride: ""
25 global:
26   imagePullSecrets: []
27   imageRegistry: ""
28   storageClass: accordion-storage
29 image:
30   debug: false

```

상태

```

1 config:
2   architecture: standalone
3   auth:
4     createDatabase: true
5     customPasswordFiles: {}
6     database: petstore
7     defaultAuthenticationPlugin: ""
8     existingSecret: ""
9     password: admin
10    replicationPassword: ""
11    replicationUser: replicator
12    rootPassword: root
13    usePasswordFiles: false
14    username: admin
15 clusterDomain: cluster.local
16 commonAnnotations: {}
17 commonLabels: {}
18 diagnosticMode:
19   args:
20     - infinity
21   command:
22     - sleep
23   enabled: false
24 extraDeploy: []
25 fullnameOverride: ""
26 global:
27   imagePullSecrets: []
28   imageRegistry: ""
29   storageClass: accordion-storage
30 image:

```

상태 탭에서는 설정값과 배포 상태 정보를 제공한다.

mysql

Resources Status History

차트들(들) 선택하세요. ^

업그레이드

- cloudnativeapp/mysql
- bitnami/mysql

```

1 architecture: standalone
2 auth:
3   createDatabase: true
4   customPasswordFiles: {}
5   database: petstore
6   defaultAuthenticationPlugin: ""
7   existingSecret: ""
8   password: admin
9   replicationPassword: ""
10  replicationUser: replicator
11  rootPassword: root
12  usePasswordFiles: false
13  username: admin
14  clusterDomain: cluster.local
15  commonAnnotations: {}
16  commonLabels: {}
17  diagnosticMode:
18    args:
19      - infinity
20    command:
21      - sleep
22    enabled: false
23  extraDeploy: []
24  fullnameOverride: ""
25  global:
26    imagePullSecrets: []
27    imageRegistry: ""
28    storageClass: accordion-storage
29  image:
30    debug: false
  
```

```

1 config:
2   architecture: standalone
3   auth:
4     createDatabase: true
5     customPasswordFiles: {}
6     database: petstore
7     defaultAuthenticationPlugin: ""
8     existingSecret: ""
9     password: admin
10    replicationPassword: ""
11    replicationUser: replicator
12    rootPassword: root
13    usePasswordFiles: false
14    username: admin
15    clusterDomain: cluster.local
16    commonAnnotations: {}
17    commonLabels: {}
18    diagnosticMode:
19      args:
20        - infinity
21      command:
22        - sleep
23      enabled: false
24    extraDeploy: []
25    fullnameOverride: ""
26    global:
27      imagePullSecrets: []
28      imageRegistry: ""
29      storageClass: accordion-storage
30    image:
  
```

우측 상단의 차트 선택 후 업그레이드 버튼을 클릭해 배포된 애플리케이션을 업그레이드할 수 있다.

mysql

Resources Status History

계정 ↑	업데이트	상태	차트	앱 버전	물백
1	2023-11-09 14:22:06	deployed	mysql-9.14.1	8.0.35	물백

히스토리 탭에서는 애플리케이션의 업데이트 내역 등에 대한 정보와 특정 업데이트 내역으로 회귀할 수 있는 기능을 제공한다.

애플리케이션 삭제

배포된 애플리케이션을 삭제하기 위해서는 대상 애플리케이션의 우측에 있는 삭제 버튼을 선택한다.

애플리케이션 > 애플 > 애플 탭

이름 ↓	네임 스페이스	계정	업데이트	상태	차트	앱 버전	삭제
test-aggr	huu	1	2024-03-06 14:33:55	deployed	nginx-15.1.2	1.25.1	삭제
test	default	1	2024-03-26 15:04:22	deployed	nginx-15.1.2	1.25.1	삭제
test	huu	5	2024-03-05 10:13:35	deployed	nginx-15.4.0	1.25.3	삭제
spin-containerd-shim-installer	hwpa	1	2024-03-06 14:33:55	deployed	spin-containerd-shim-installer-0.10.0	0.10.0	삭제
opensearch-dashboards	alert	1	2024-03-06 14:33:55	deployed	opensearch-dashboards-1.3.0	1.3.0	삭제
opensearch	alert	1	2024-03-06 14:33:55	deployed	opensearch-1.9.0	1.3.0	삭제
mysql	mjdev	1	2024-01-04 10:04:16	deployed	mysql-9.14.1	8.0.35	삭제
mysql	alert	1	2023-11-09 14:22:06	deployed	mysql-9.14.1	8.0.35	삭제
locust	ingresstest	5	2024-07-02 22:23:59	deployed	locust-0.31.5	2.15.1	삭제
locust	stress	1	2024-06-27 14:37:24	deployed	locust-0.31.5	2.15.1	삭제

삭제

애플리케이션(들) 삭제하시겠습니까?

취소 **삭제**

이전 1 2 다음

4.2.5. 빌드

4.2.5.1. 클러스터 태스크 템플릿

클러스터 태스크 템플릿에서는 클러스터에서 공통으로 사용할 템플릿을 관리한다.

파이프라인에서 자주 사용되는 태스크 사양을 미리 템플릿으로 등록하여 쉽게 적용할 수 있다.

태스크 템플릿을 만들 때는 **valueschema** 와 **환경변수** 를 사용한다.

valueschema 는 사용자에게 입력받을 값(value)에 대한 형식의 정의할 수 있으며, 템플릿을 유연하게 만들 수 있다.

데이터 형식에 따라 크게 3가지 유형으로 세부 타입은 아래와 같다.

- **기본** : 리터럴 데이터
- **쿠버네티스** : 쿠버네티스 리소스
- **참조** : 다른 스키마를 포함

유형	종류	설명
기본	boolean	true, false
기본	number	숫자 형식
기본	string	문자열 형식
쿠버네티스	kubernetes	범용적으로 사용할 수 있는 것으로 쿠버네티스 리소스를 참조
쿠버네티스	configmapName	configmap 존재 여부만 확인하며, metadata.name을 참조
쿠버네티스	configmapData	configmap의 data의 요소를 참조
쿠버네티스	secretName	secret 존재 여부만 확인하며, metadata.name을 참조
쿠버네티스	secretData	secret의 data의 요소를 참조
참조	array	동일한 유형의 객체를 포함
참조	object	다른 객체를 포함하는 유형
참조	anyOf	유형 중에 하나 일치

환경변수

파이프라인에서 예약된 환경변수는 다음과 같다. 사용 방법은 `acc-showcase-env` 파이프라인 템플릿 참고한다.

의미	환경변수 표현식
파이프라인 이름	{{{.PIPELINE.NAME}}}
파이프라인 네임스페이스	{{{.PIPELINE.NAMESPACE}}}
파이프라인 UID	{{{.PIPELINE.UID}}}

의미	환경변수 표현식
파이프라인 인스턴스 이름	{{{.PIPELINE.INSTANCE}}}
빌드 이름	{{{.BUILD.NAME}}}
빌드 버전	{{{.BUILD.VERSION}}}
빌드 생성자 이름	{{{.BUILD.CREATOR.USERNAME}}}
빌드 생성자 그룹	{{{.BUILD.CREATOR.GROUPS}}}

valueschema의 기본 유형

boolean

입력값을 불리언 형식으로 제한한다.

```
spec:
  container:
    image: busybox
    command:
      - sh
      - -c
    args:
      - |
        echo -e "bVal: {{.values.bVal}}";
        echo -e "bDefaultVal: {{.values.bDefaultVal}}";
        echo -e "bEnumVal: {{.values.bEnumVal}}";
  valueschema:
    type: object
    properties:
      bVal:
        type: boolean
        description: "기본형: boolean타입"
      bDefaultVal:
        type: boolean
        description: "기본값이 있는 boolean타입"
        default: true
      bEnumVal:
        type: boolean
        description: "boolean 타입, enum : 값의 범위 지정"
        enum:
          - true
```

YAML

bDefaultVal

기본값이 있는 boolean타입

bEnumVal

boolean 타입, enum : 값의 범위 지정

bVal

기본형: boolean타입

number

입력값을 숫자 형식으로 제한한다.

YAML

```
spec:
  container:
    image: busybox
    command:
      - sh
      - -c
    args:
      - |
        echo -e "nVal: {{.values.nVal}}";
        echo -e "nDefaultVal: {{.values.nDefaultVal}}";
  valueschema:
    type: object
    properties:
      nVal:
        type: number
        description: "기본형: number 타입"
        maximum: 100 # (옵션) 해당 값보다 크다면 오류
        minimum: 1 # (옵션) 해당 값보다 작다면 오류
      nDefaultVal:
        type: number
        description: "기본값이 있는 number 타입"
        default: -1
```

nDefaultVal

기본값이 있는 number 타입

nVal

기본형: number 타입

string

입력값을 문자열 형식으로 정의하고, 다양한 포맷을 이용하여 제한할 수 있다.

종류	설명
date	날짜형식 (YYYY-MM-DD)
email	이메일 형식
ip	ipv4 형식
uri	uri 형식
uuid	uuid 형식
ssh-private	SSH PEM PRIVATE 형식
ssh-public	ssh-rsa xxxx accordion@example.com
textarea	일반적인 형식의 멀티라인

```
spec:
  container:
    image: busybox
    command:
      - sh
      - -c
    args:
      - |
        echo -e "date : {{.values.date}}";
        echo -e "email : {{.values.email}}";
        echo -e "ip : {{.values.ip}}";
        echo -e "uri : {{.values.uri}}";
        echo -e "uuid : {{.values.uuid}}";
        echo -e "textarea : {{.values.textarea}}";
        echo -e "sshPublic : {{.values.sshPublic}}";
  valueschema:
    type: object
    properties:
      date:
        type: string
        description: "date 형식 (YYYY-MM-DD)"
        format: date
      email:
        type: string
        description: "email 형식"
        format: email
      ip:
        type: string
        description: "ip 형식"
        format: ip
      uri:
        type: string
        description: "uri 형식"
        format: uri
      uuid:
        type: string
        description: "uuid 형식"
        format: uuid
      textarea:
        type: string
        description: "멀티라인 형식 지원"
        format: textarea
      sshPublic:
        type: string
        description: "ssh public 형식"
        format: ssh-public
```

date

date 형식 (YYYY-MM-DD)

email

email 형식

ip

ip 형식

sshPublic

ssh public 형식

textarea

멀티라인 형식 지원

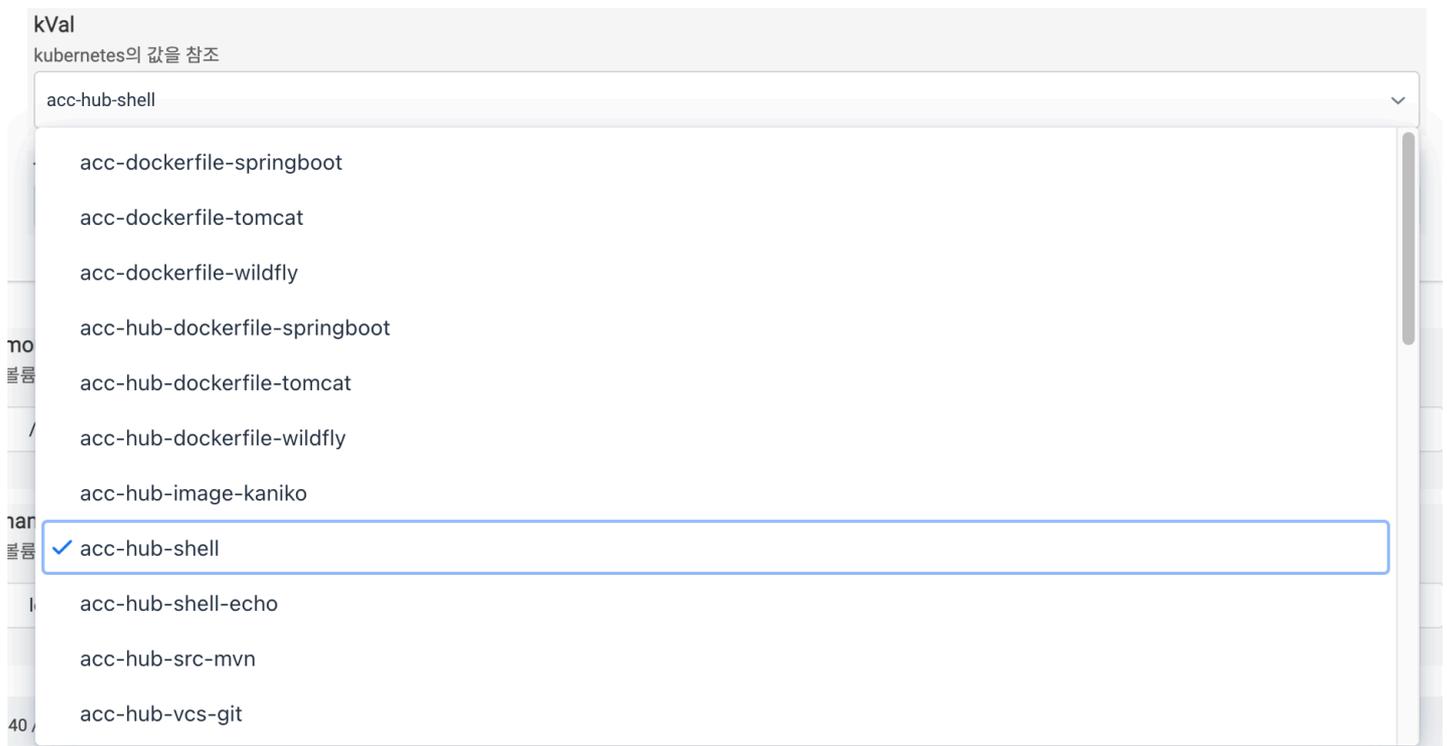
valueschema의 쿠버네티스 유형

kubernetes

입력값을 쿠버네티스 리소스를 선택하도록 한다.

YAML

```
spec:
  container:
    image: busybox
    command:
      - sh
      - -c
    args:
      - |
        echo -e "kVal: {{.values.kVal}}";
  valueschema:
    type: object
    properties:
      kVal:
        type: kubernetes
        description: "kubernetes의 값을 참조"
        x-kube-group: "cicd.accordions.co.kr" # (필수) 쿠버네티스 리소스의 그룹 의미
        x-kube-version: "v1beta1" # (필수) 쿠버네티스 리소스의 버전 의미
        x-kube-resource: "tasktemplates" # (필수) 쿠버네티스 리소스의 리소스명 의미
        x-kube-namespace: "namespace 이름" # (옵션) 쿠버네티스 리소스의 네임스페이스 의미
        x-kube-fields: "metadata.name" # (옵션) jsonpath 유사한 형식으로 "{}" 는 생략 가능
        x-kube-optional: true # (옵션) true이면서 사용자 입력값이 없다면 공백("")
```



configmapData

입력값을 configmap 리소스를 선택하도록 하며, data 필드를 참조한다.

YAML

```
spec:
  container:
    image: busybox
    command:
      - sh
      - -c
    args:
      - |
        echo -e "cmData: {{.values.cmVal}}";
  valueschema:
    type: object
    properties:
      cmVal:
        type: configmapData
        description: "configmap.data의 값을 참조"
        x-kube-key: "strkey" # (옵션) 해당값이 없다면, data전체를 json직렬화
        x-kube-optional: true # (옵션) true이면서 사용자 입력값이 없다면 공백("")
```

cmVal 생성

configmap.data의 값을 참조

- istio-ca-root-cert
- kube-root-ca.crt

configmapName

입력값을 configmap 리소스를 선택하도록 하며, metadata.name 값을 참조한다.

YAML

```
spec:
  container:
    image: busybox
    command:
      - sh
      - -c
    args:
      - |
        echo -e "cmName: {{.values.cmNameVal}}";
  valueschema:
    type: object
    properties:
      cmNameVal:
        type: configmapName
        description: "configmap의 이름을 참조"
        x-kube-labelSelector: "" # (옵션) 라벨 선택터
        x-kube-optional: true # (옵션) true이면서 사용자 입력값이 없다면 공백("")
```

cmNameVal

생성

configmap의 이름을 참조

- istio-ca-root-cert
- kube-root-ca.crt

secretData

입력값을 secret 리소스를 선택하도록 하며, data 필드의 디코딩 값을 참조한다.

YAML

```
spec:
  container:
    image: busybox
    command:
      - sh
      - -c
    args:
      - |
        echo -e "secretData: {{.values.secretVal}}";
  valueschema:
    type: object
    properties:
      secretVal:
        type: secretData
        description: "secret.data의 값을 참조"
        x-kube-key: "username" # (옵션) 해당값이 없다면, data전체를 json직렬화
        x-kube-type: "kubernetes.io/basic-auth" # (옵션) ssecret.type과 일치하는지 유효성 검사
        x-kube-labelSelector: "" # (옵션) 라벨 셀렉터
        x-kube-optional: true # (옵션) true이면서 사용자 입력값이 없다면 공백("")
```

secretVal

생성

secret.data의 값을 참조

dev-server-auth
qa-server-auth
test-server-auth

secretName

입력값을 secret 리소스를 선택하도록 하며, metadata.name 값을 참조한다.

YAML

```
spec:
  container:
    image: busybox
    command:
      - sh
      - -c
    args:
      - |
        echo -e "secretName: {{.values.secretNameVal}}";
  valueschema:
    type: object
    properties:
      secretNameVal:
        type: secretName
        description: "secret의 이름을 참조"
        x-kube-type: "kubernetes.io/basic-auth" # (옵션) ssecret.type과 일치하는지 유효성 검사
        x-kube-labelSelector: "" # (옵션) 라벨 선택터
        x-kube-optional: true # (옵션) true이면서 사용자 입력값이 없다면 공백("")
```

secretNameVal

생성

secret의 이름을 참조

- dev-server-auth
- qa-server-auth
- test-server-auth

valueschema의 참조 유형

array

입력값을 배열 형식으로 받는다.

YAML

```
spec:
  container:
    image: busybox
    command:
      - sh
      - -c
    args:
      - |
        echo -e "numArr: {{.values.numArr}}";
        echo -e "strArr: {{.values.strArr}}";
        {{- range $i, $e:= .values.strArr}}
        {{print "echo -e W"telement[" $i "]: " $e "W";}}
        {{- end}}
        echo -e "uniqueArr: {{.values.uniqueArr}}";
  valueschema:
    type: object
    properties:
      numArr:
        type: array
        description: "숫자 배열"
        items:
          type: number
      strArr:
        type: array
        minItems: 1
        maxItems: 3
        description: "문자 배열 (1~3)"
        items:
          type: string
      uniqueArr:
        type: array
        description: "중복 요소가 없는 배열"
        uniqueItems: true
        items:
          type: string
```

numArr

숫자 배열

추가

strArr

문자 배열 (1~3)

삭제

1

idx-1

삭제

2

idx-2

추가

uniqueArr

중복 요소가 없는 배열

추가

object

입력값을 오브젝트 형식으로 받는다.

YAML

```
spec:
  container:
    image: busybox
    command:
      - sh
      - -c
    args:
      - |
        echo -e "data.key: {{.values.data.key}}";
        echo -e "data.value: {{.values.data.value}}";
  valueschema:
    type: object
    properties:
      data:
        type: object
        properties:
          key:
            type: string
          value:
            type: string
```

data

key

config

value

application.yaml

anyOf

입력값을 두 객체 중에 하나를 선택하는 방식으로 받는다.

YAML

```
spec:
  container:
    image: busybox
    command:
      - sh
      - -c
    args:
      - |
        {{- if eq .values.secret.type "kubernetes.io/basic-auth"}}
        {{print "echo W" .values.secret.type "W";}}
        {{print "echo W" .values.secret.username "W";}}
        {{print "echo W" .values.secret.password "W";}}

        {{- else if eq .values.secret.type "kubernetes.io/ssh-auth"}}
        {{print "echo W" .values.secret.type "W";}}
        {{print "echo W" .values.secret.privatekey "W";}}
        {{- end}}
  valueschema:
    type: object
    properties:
      secret:
        anyOf:
          - title: basic
            properties:
              type:
                type: string
                enum:
                  - kubernetes.io/basic-auth
              username:
                type: string
              password:
                type: string
          - title: ssh
            properties:
              type:
                type: string
                enum:
                  - kubernetes.io/ssh-auth
            privatekey:
              type: string
              format: ssh-private
```

secret

basic	▼
basic	
ssh	

type

kubernetes.io/basic-auth	▼
--------------------------	---

username

--

valueschema의 UI 옵션 정의

x-ui-format

UI에 대한 표현 방식을 정의한다. (기본값: list)

- list
 - 기본 목록 방식으로 각 properties를 행마다 표시한다.

포트 설정 (Ports) ?

포트 설정 1

containerPort

name

nodePort

protocol

servicePort

삭제

포트 설정 추가

- table
 - 테이블 방식으로 각 properties를 열마다 표시한다.
 - 세부 옵션
 - x-ui-table-width
 - x-ui-format: table 인 경우에만 옵션 적용된다.
 - 각 숫자는 컬럼의 비율을 의미한다.
 - properties 개수와 일치하지 않으면 해당 옵션 필드는 무시된다.

사용 예시 1: 포트 설정

```

ports:
  description: 포트 설정
  items:
    properties:
      containerPort:
        maximum: 65535
        minimum: 1
        multipleOf: 1
        type: number
      name:
        default: ""
        type: string
      nodePort:
        default: 0
        maximum: 65535
        minimum: 0
        multipleOf: 1
        type: number
      protocol:
        default: TCP
        enum:
          - TCP
          - UDP
          - SCTP
        type: string
      servicePort:
        maximum: 65535
        minimum: 1
        multipleOf: 1
        type: number
    type: object
  type: array
  x-ui-format: table
  x-ui-table-width: 4 2 2 2 2

```

포트 설정 (Ports) ?

containerPort	name	nodePort	protocol	servicePort
<input type="text" value="8080"/>	<input type="text" value="http-port"/>	<input type="text" value="0"/>	<input type="text" value="TCP"/>	<input type="text" value="8080"/>

[포트 설정 추가](#)

위처럼 각 property 가 한 행으로 지정한 비율에 맞추어 표시된다.

사용 예시 2: Labels 설정

YAML

```

labels:
  items:
    properties:
      key:
        pattern: ([A-Za-z0-9][-A-Za-z0-9_.]*)?[A-Za-z0-9]
        type: string
      value:
        default: ""
        pattern: ((([A-Za-z0-9][-A-Za-z0-9_.]*)?[A-Za-z0-9])?)
        type: string
    type: object
  x-ui-displayName: 라벨
  type: array
  x-ui-basic-option: true
  x-ui-displayName: 라벨
  x-ui-format: table

```

라벨

key	value
<input type="text"/>	<input type="text"/>

[라벨 추가](#)

자주 사용되는 key-value 형태의 UI 형식으로 labels, annotations 등에 사용된다.

- count
 - type: number 인 경우 사용 가능한 옵션으로, 해당 필드의 숫자를 변경할 수 있는 UI를 제공한다.

사용 예시: Replicas 설정

YAML

```

replicas:
  default: 1
  minimum: 0
  type: number
  x-ui-format: count
  x-ui-displayName: 레플리카 파드 수

```

레플리카 파드 수

- switch
 - type: boolean 인 경우 사용 가능한 옵션으로, 해당 필드의 boolean 값을 변경할 수 있는 UI를 제공한다.

사용 예시: ReadOnly 설정

YAML

```

readOnly:
  default: false
  type: boolean
  x-ui-format: switch

```

readOnly



• radio

- type: string 과 enum 이 존재 하는 경우 사용 가능한 옵션으로, enum 목록으로 정의된 값을 선택할 수 있는 UI를 제공한다.

사용 예시: ImagePullPolicy 설정

YAML

```
imagePullPolicy:
  default: Always
  description: 이미지를 가져오는 전략
  enum:
    - Always
    - IfNotPresent
    - Never
  type: string
  x-ui-format: radio
```

imagePullPolicy ?

- Always
- IfNotPresent
- Never

x-ui-collapsible

Collapse, Expand 버튼이 표시되어, 해당 기본값을 지정할 수 있는 옵션을 정의한다.

- true: 숨김
- false: 표시

사용 예시

YAML

```
ports:
  ...
  x-ui-collapsible: true
```

> ports

포트 설정

x-ui-order

기본은 알파벳순으로 정렬이 되며, 사용자가 UI 배치 변경을 위한 옵션을 정의한다.

사용 예시

YAML

```
propertyA:
  type: string
  x-ui-order: 2
propertyB:
  type: string
  x-ui-order: 1
```

현재 태스크 파라미터

propertyB

propertyA

위처럼 알파벳순이 아닌 사용자가 정의한 순서대로 UI가 표시된다.

x-ui-basic-option

카탈로그 생성시 기본으로 표시되는 UI를 정의한다.

- true: 생성시, 기본 설정에서 표시
- false: 생성시, 기본 설정에서 미표시 (전체 설정에서만 표시)

사용 예시

YAML

```
ports:
  ...
  x-ui-basic-option: true
```

x-ui-displayName

key 값이 표시되는 UI의 이름을 정의한다.

사용 예시

YAML

```
ports:
  ...
  x-ui-displayName: 포트 설정
```

x-ui-display

UI 표시/숨김을 위한 옵션을 정의한다.

- visible: UI를 표시한다.
- hidden: UI를 숨김 처리한다.

x-ui-type: kubernetes

type: string 인 경우만 지원한다. 사용자의 특정 쿠버네티스 목록을 UI로 제공하기 위한 옵션이다.

NOTE

type: kubernetes 에서 사용되는 x-kube- 와 다른 옵션이다. x-ui-type- 옵션의 경우는 CICD 모듈과 관계되지 않고 사용자 UI에만 영향 미치는 옵션이다.

세부 옵션

옵션	설명
x-ui-kube-resource	리소스명
x-ui-kube-group	리소스 그룹 (기본값: core) - 예시: apps , storage.k8s.io
x-ui-kube-version	리소스 버전 (기본값: v1) - 예시: v1beta1 , v2 , v2beta2
x-ui-kube-scope	쿠버네티스 리소스의 스코프(cluster, namespace)를 정의한다. - cluster : 클러스터 스코프의 리소스 - namespace : 네임스페이스 스코프의 리소스 (기본값)
x-ui-kube-fields	(필수값) UI 목록으로 표시할 리소스의 참조 필드 - 기본적으로 metadata.name 을 정의하는 것을 권장한다.
x-ui-kube-labelSelector	리소스 목록 요청시 전달되는 라벨 셀렉터

사용예시

RESOURCE YAML

YAML

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: {{{.CATALOG.NAME}}}
spec:
  accessModes:
  - {{.values.accessMode}}
  resources:
    requests:
      storage: {{.values.size}}
  storageClassName: {{.values.storageClass}}
  volumeMode: Filesystem

```

VALUESchema YAML

```

properties:
  accessMode:
    type: string
  size:
    type: string
  storageClass:
    description: 스토리지 클래스
    type: string
    x-ui-kube-fields: metadata.name
    x-ui-kube-group: storage.k8s.io
    x-ui-kube-labelSelector: ''
    x-ui-kube-resource: storageclasses
    x-ui-kube-scope: cluster
    x-ui-kube-version: v1
    x-ui-type: kubernetes
type: object

```

입력값

accessMode

ReadWriteOnce

size

1

✓ accordion-storage

accordion-storage

위처럼 사용가능한 스토리지 클래스 목록이 표시된다.

4.2.5.2. 클러스터 파이프라인 템플릿

클러스터 파이프라인 템플릿에서는 클러스터에서 공통으로 사용할 파이프라인을 관리한다.

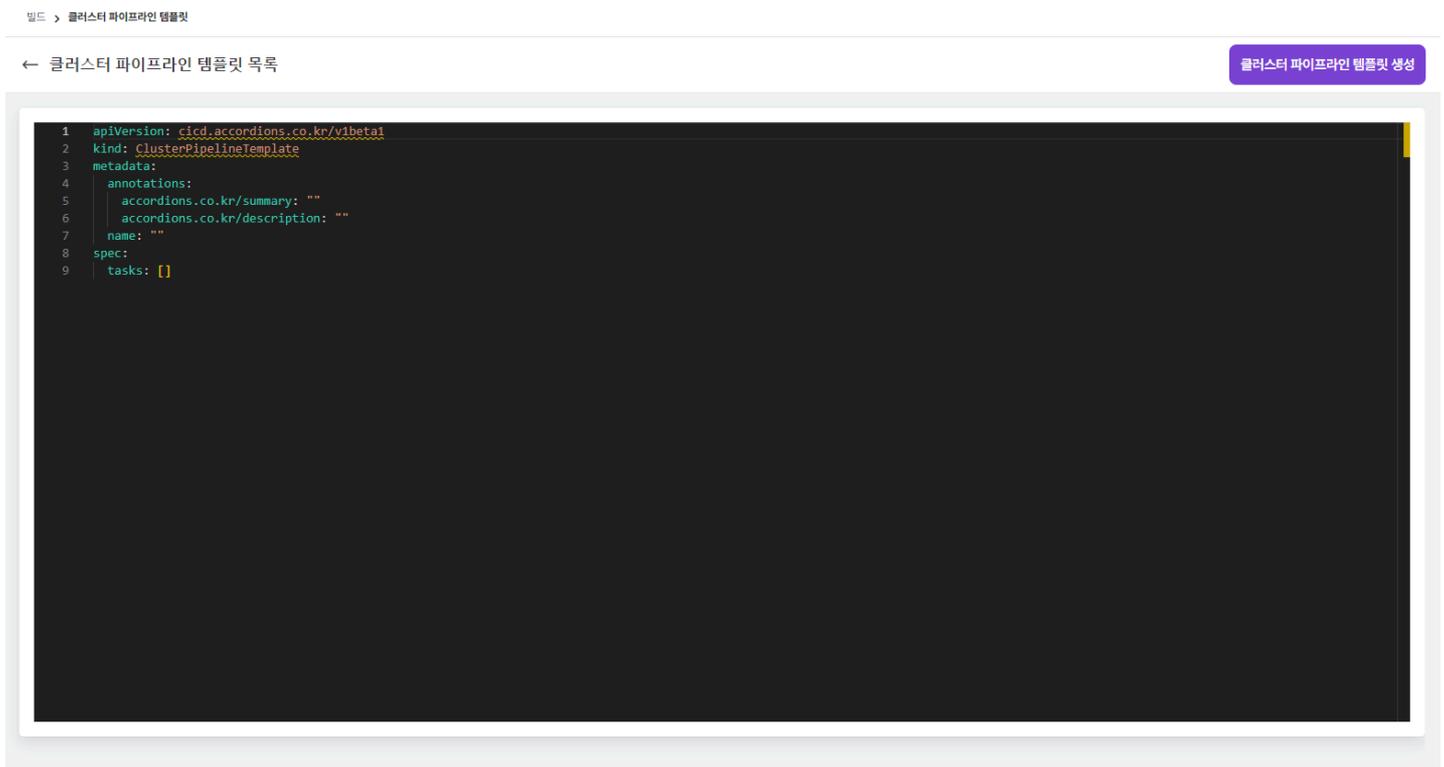
카탈로그에서 자주 사용되는 파이프라인 사양을 미리 템플릿으로 등록하여 쉽게 적용할 수 있다.

클러스터 파이프라인 템플릿을 만들 때는 클러스터 태스크 템플릿을 사용하며

2개 이상의 클러스터 태스크 템플릿을 조합하는 경우 클러스터 태스크 템플릿들의 연관관계 설정이 필요하다.

클러스터 파이프라인 템플릿 생성

+ 클러스터 파이프라인 템플릿 생성 버튼을 선택하면 나타나는 화면에서 클러스터 파이프라인 템플릿 yaml 정보를 입력하여 생성할 수 있다.



```

apiVersion: cicd.accordions.co.kr/v1beta1
kind: ClusterPipelineTemplate
metadata:
  annotations:
    # 해당 클러스터 파이프라인 템플릿의 요약
    accordions.co.kr/summary: ""
    # 해당 클러스터 파이프라인 템플릿의 설명
    accordions.co.kr/description: ""
  name: "test01"
spec:
  tasks:
    - name: vcs-get
      templateRef:
        clusterScope: true
        name: acc-vcs-git
    - depends: vcs-get.Succeeded
      name: image-build
      templateRef:
        clusterScope: true
        name: acc-image-kaniko

```

YAML

태스크에 대한 스펙을 작성할때에는 이름과 다른 태스크와의 연관관계를 설정하고 상세정보는 태스크 템플릿을 기반으로 작성한다.

태스크의 이름은 쿠버네티스 이름 정책에 맞춰 작성하고 태스크의 연관관계는 해당 태스크를 수행하는 조건에 대한 정보를 입력한다. 이는 {이름}. {상태} 의 형식으로 입력할 수 있다. 상태에 입력할 수 있는 값은 다음과 같다.

상태	설명
Terminated	사용자가 빌드를 중지한 상태
Succeeded	태스크가 수행하고 종료코드(exit code)가 0으로 완료된 상태
Running	태스크가 수행되고 있는 상태
Failed	태스크가 수행되었지만 종료코드가 0이 아닌 것으로 완료된 상태
Error	태스크가 실행되지 못하거나 비정상 종료된 상태
Pending	태스크가 수행 전 대기하는 상태
Unknown	알수 없는 문제로 종료된 상태

연관관계 작성 방법은 다음과 같다.

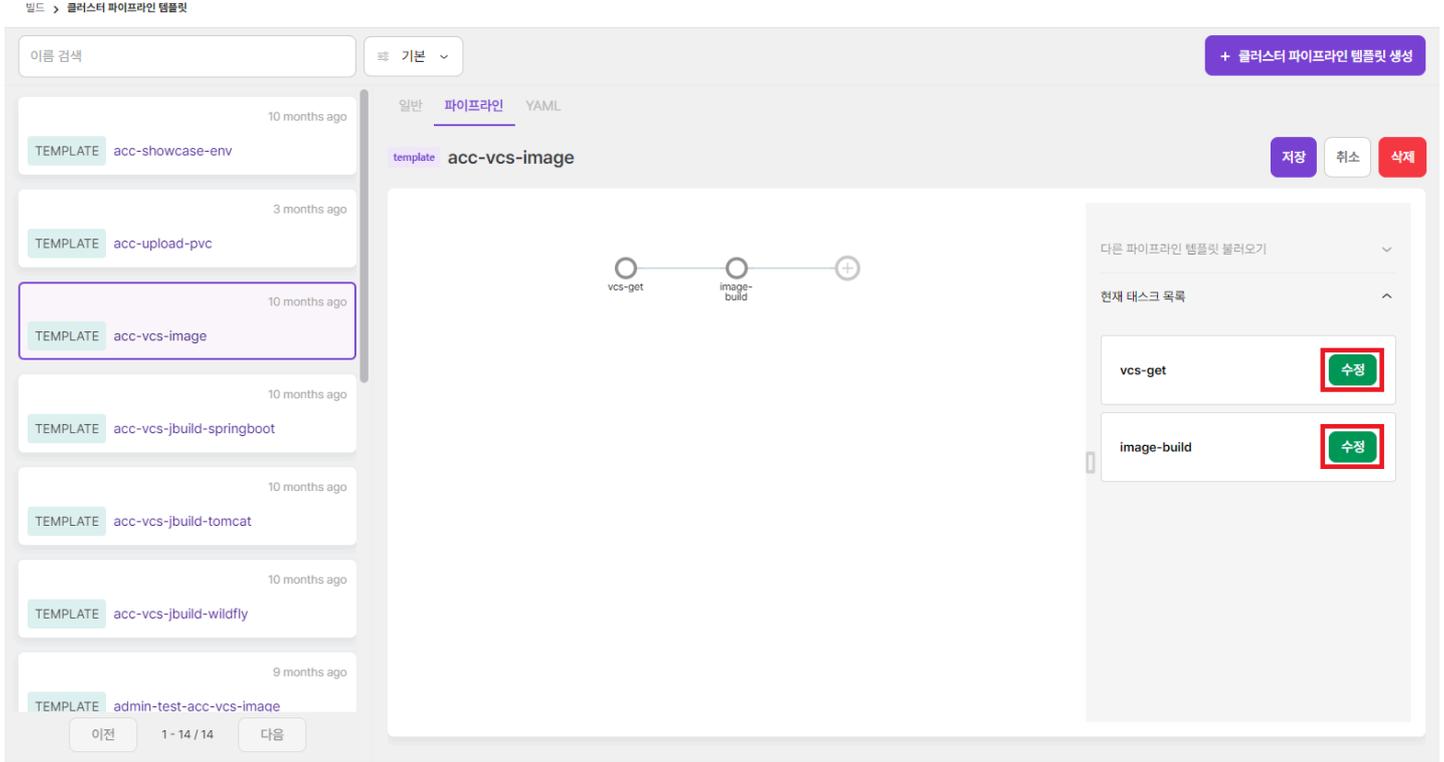
- 태스크의 {이름}.{상태} 형식으로 입력하며 상태는 필수로 입력해야 한다.
- &&, || 를 추가하여 논리적인 표현도 작성 가능하다.
 - && 는 and 를 뜻하는 표현으로 A.{상태} && B.{상태} 인 경우 A 상태와 B 상태 모두인 경우를 표현한다.
 - || 는 or 를 뜻하는 표현으로 A.{상태} || B.{상태} 인 경우 A 상태 또는 B 상태 중 하나를 표현 한다.
- ({이름}.{상태} || {이름}.{상태}) && {이름}.{상태} 와 같은 우선순위 지정도 가능하다.

```
spec:
  tasks:
    - name: vcs-get
      templateRef:
        clusterScope: true
        name: acc-vcs-git
    - name: test-task
      templateRef:
        clusterScope: true
        name: test-task-template
    - depends: (test-task.Succeeded || test-task.Running) && vcs-get.Succeeded
      name: image-build
      templateRef:
        clusterScope: true
        name: acc-image-kaniko
```

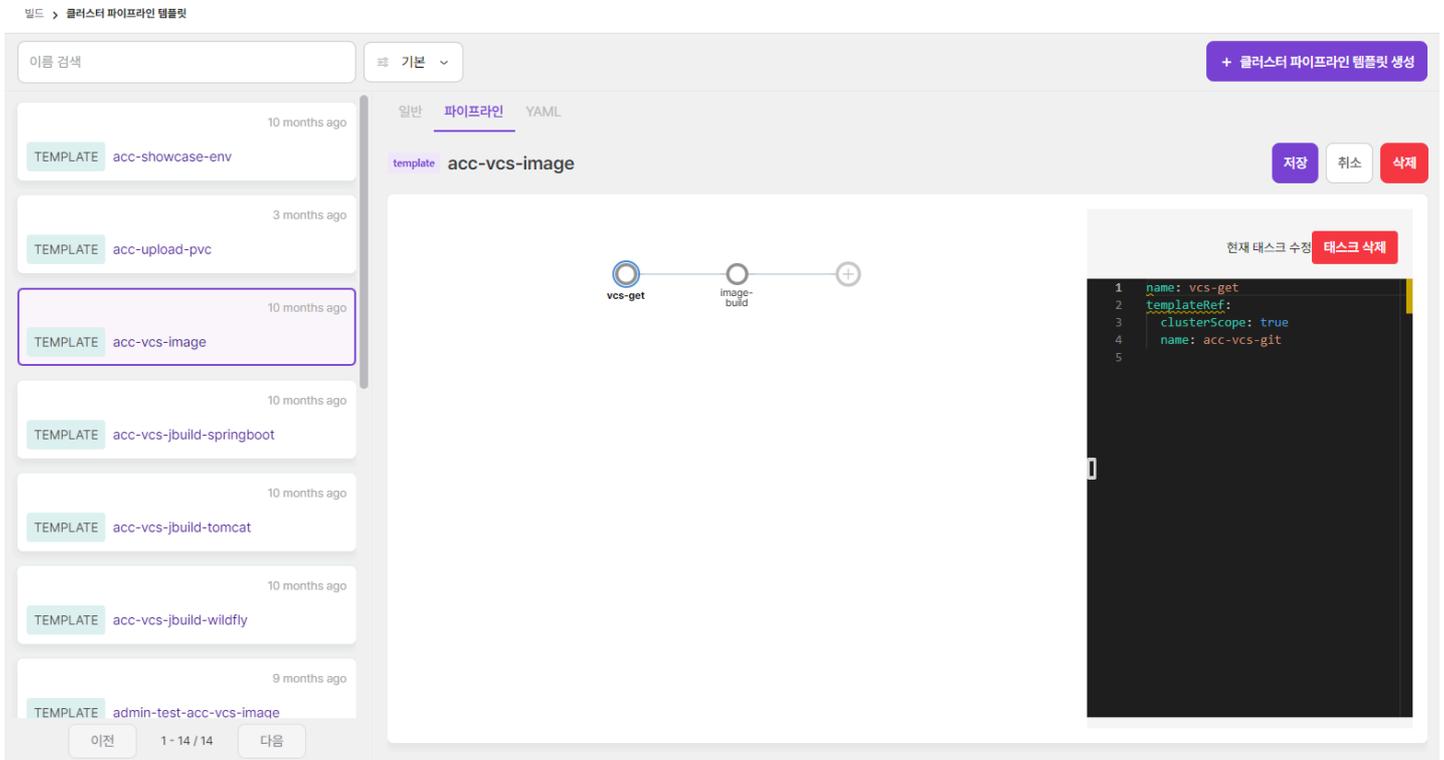
YAML

클러스터 파이프라인 템플릿 수정

클러스터 파이프라인 템플릿의 태스크에 대한 정보를 변경해야하는 경우 또는 클러스터 파이프라인 템플릿을 빈 템플릿으로 만들어 변경할 필요가 있는 경우 파이프라인 탭의 수정 버튼을 선택한다.



수정을 원하는 태스크의 수정 버튼을 선택한다.

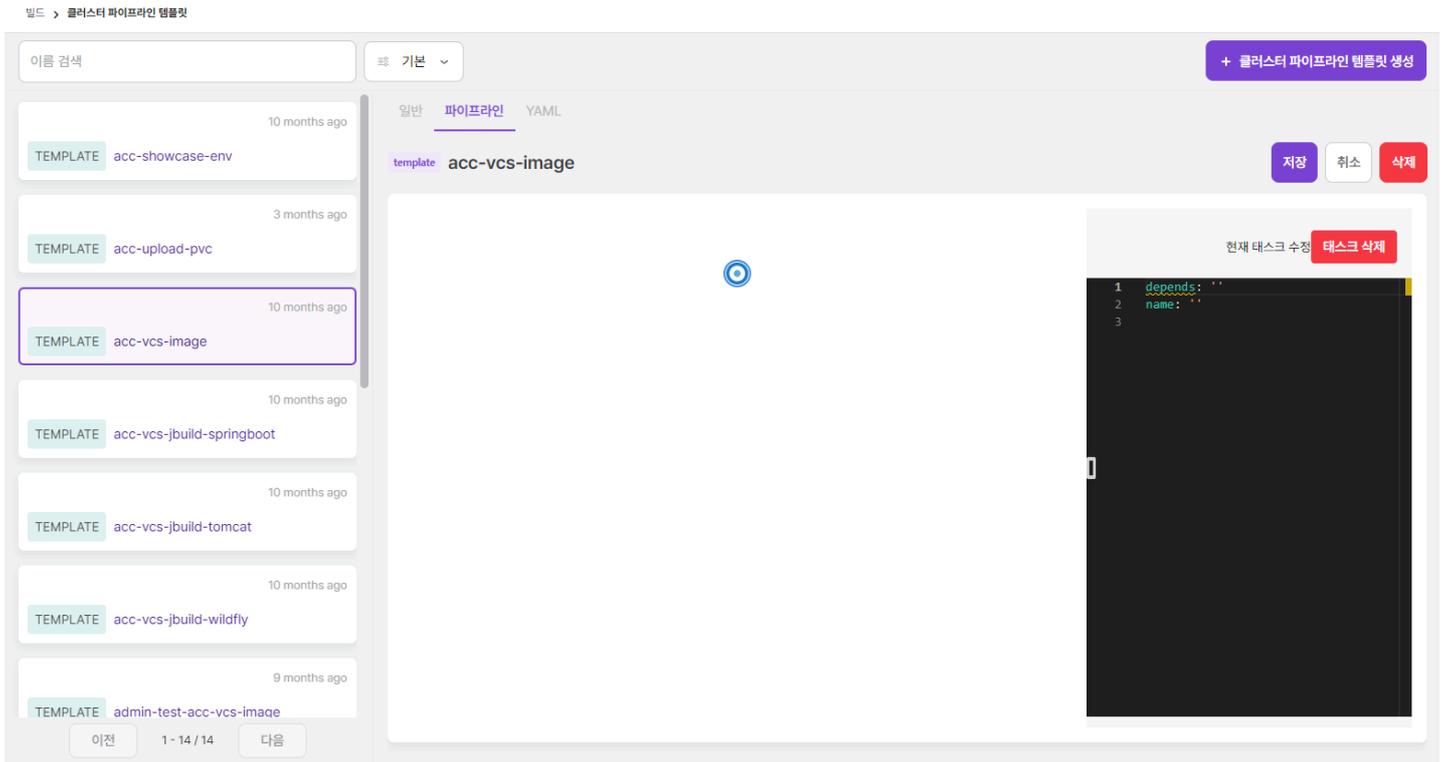


NOTE

태스크가 없는 클러스터 파이프라인 템플릿의 경우에는 수정하여, 빈 태스크의 + 버튼을 선택하면 태스크 구성을 할 수 있다.

태스크 생성

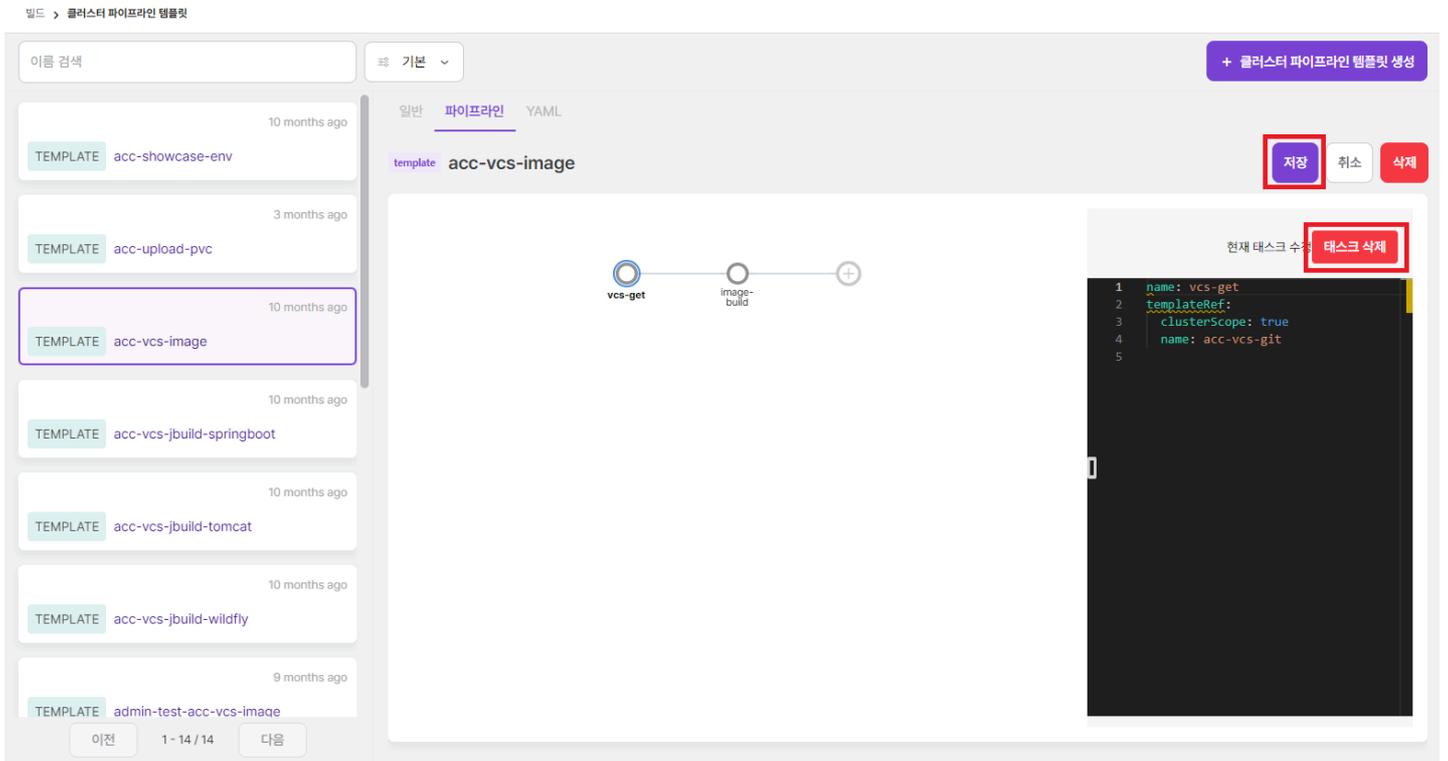
파이프라인 탭의 수정 버튼 클릭, + 버튼을 선택 후 나오는 YAML 에서 입력 한다.



태스크 삭제

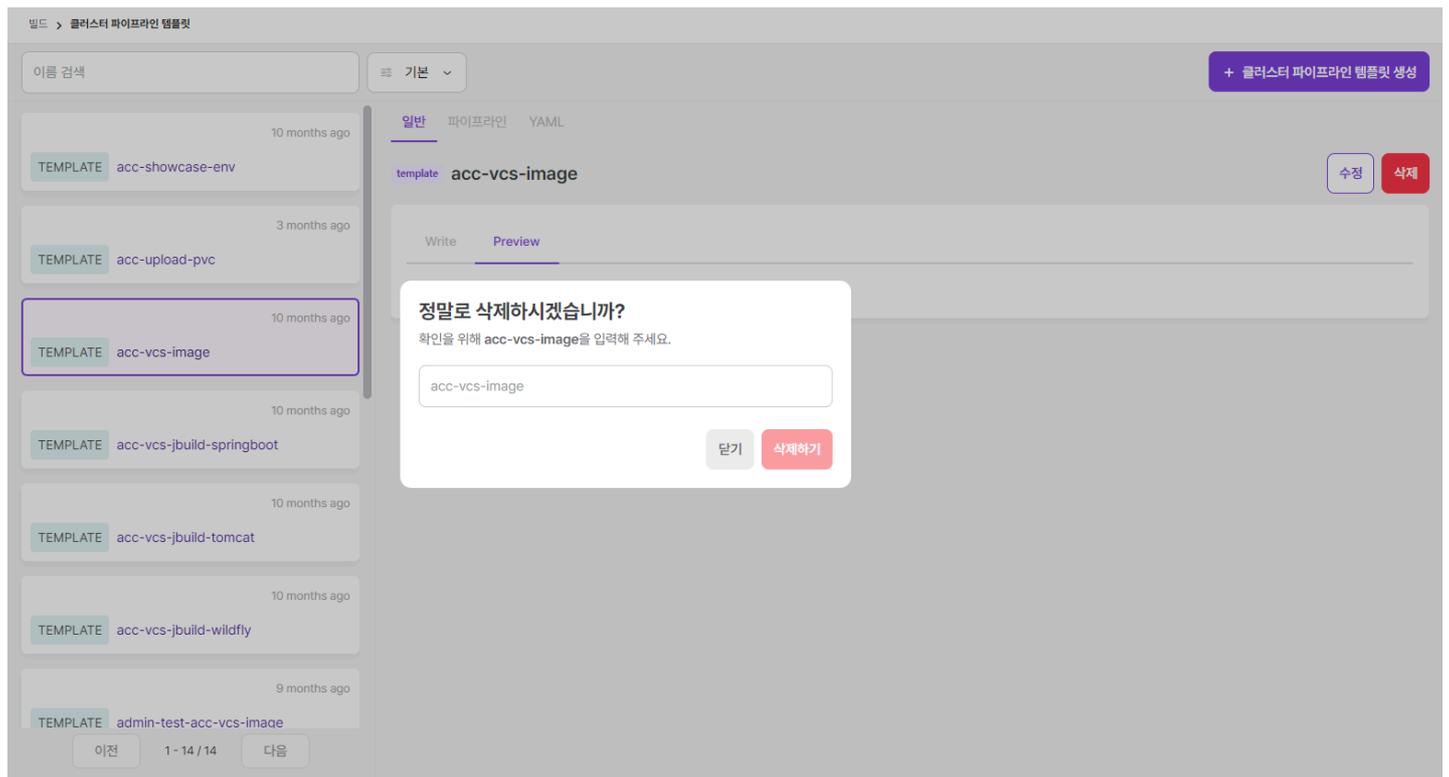
상단의 파이프라인 탭을 선택한다.

그리고 삭제를 원하는 task를 선택한 다음 수정 버튼을 누르고, 태스크 삭제 버튼을 클릭 후 우측 상단의 저장 버튼을 클릭한다.



클러스터 파이프라인 템플릿 삭제

삭제하려는 클러스터 파이프라인 템플릿을 선택하고 우측 상단의 삭제 버튼을 클릭 한다.



이후 표시되는 모달에서 이름을 입력하고 삭제하기 버튼을 선택한다.

4.2.6. 워크로드

워크로드는 쿠버네티스에서 구동되는 애플리케이션을 의미한다. 쿠버네티스에서 워크로드는 파드의 집합에서 실행되며 파드는 실행 중인 컨테이너의 집합이다. 워크로드의 종류는 아래와 같다.

- 디플로이먼트
- 스테이트풀셋
- 데몬셋
- 레플리카셋
- 잡
- 크론잡
- 파드

4.2.6.1. 워크로드 대시보드

워크로드 대시보드는 클러스터에 배포된 워크로드 상태 정보를 제공한다.

워크로드 > 워크로드 대시보드

■ Running
 ■ Pending
 ■ Succeeded
 ■ Running
 ■ NotRunning
 ■ Warning
 ■ Failed

워크로드 대시보드

워크로드 리소스에 대한 정보 조회를 실시간으로 제공합니다.

파드	280
디플로이먼트	163
스테이트풀셋	19
데몬셋	16
잡	27
크론잡	12

> 이벤트 (NORMAL 71 / WARNING 51)

차트에서 워크로드를 선택하면 해당 워크로드에 대한 상세 정보를 조회할 수 있다.

워크로드 > 워크로드 대시보드

워크로드 대시보드
워크로드 리소스에 대한 정보 조회를 실시간으로 제공합니다.

종류	-
이름	blackbox-exporter-5695fc478b-s5mdl
네임스페이스	acc-system
상태	Running
Ready	3/3
생성 시간	2024-07-05 22:52:09
이벤트	
메시지 ↑	시간

데이터가 존재하지 않습니다.

> 이벤트 (NORMAL 71 / WARNING 51)

파드의 경우 상태가 Running 이지만, 정상적으로 동작하고 있지 않는 컨테이너가 존재하는 경우 해당 파드의 컨테이너 목록을 보여준다.

워크로드 > 워크로드 대시보드

워크로드 대시보드
워크로드 리소스에 대한 정보 조회를 실시간으로 제공합니다.

종류	Pod	
이름	test-aggr-nginx-687bb678b7-8dxdpq	
네임스페이스	huu	
상태	Running	
Ready	0/1	
컨테이너 목록		
이름 ↑	상태	원인
nginx	● waiting	CrashLoopBackOff
생성 시간	2024-09-10 17:01:06	
이벤트		
메시지 ↑	시간	

Back-off restarting failed container nginx in pod test-aggr-nginx-687bb678b7-8dxdpq_huu(eaf08199-756d-418d-9b73-c055b5c22661)

> 이벤트 (NORMAL 72 / WARNING 51)

4.2.6.2. 파드

배포된 파드 정보를 제공한다.

탭을 이용해 파드에 대한 쿠버네티스 리소스 정보, 컨테이너 로그 조회 및 터미널 접속, 토폴로지 조회가 가능하다. 컨테이너 로그 조회와 터미널 접속은 파드의 컨테이너 별로 가능하다.

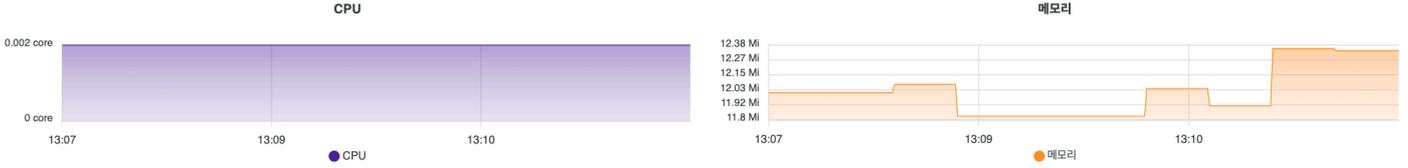
개요

배포된 파드의 메트릭, 컨테이너, 이벤트, 상태, 세부 정보를 제공한다.

The screenshot displays the 'Overview' page for a pod named 'nginx-1-58d57c9f8-l8wrp'. The interface includes a search bar, navigation tabs (개요, 로그, 터미널, 토폴로지, YAML), and a '+ 파드 생성' button. The pod's status is 'Running' with 1/1 ready containers. Below this, there are two charts: 'CPU' usage (1 core) and '메모리' (Memory) usage (18 Mi). A table lists the container 'nginx-1' with its state as 'running', image as 'nginx', and memory usage as 12.441 Mi. An '이벤트' section shows no events. At the bottom, a '컨디션' section lists pod labels and host IP (10.20.200.204).

메트릭 정보

메트릭 정보는 현재 시간으로부터 5분간 CPU, Memory 사용량을 제공한다.



컨테이너 정보

배포된 파드의 컨테이너에 대한 정보를 제공한다. 컨테이너 테이블 열을 선택하면 Arguments, Env, Volume에 대한 정보를 조회할 수 있다.

The screenshot shows a container management interface for a pod named 'nfs-client-provisioner'. The container is in a 'running' state. Key details include:

- Env:** PROVISIONER_NAME: accordion-data-provisioner, NFS_SERVER: 10.10.0.84, NFS_PATH: /volume1/dev/host-cluster
- Volume:** A table listing volumes: 'kube-api-access-4w84l' (path: /var/run/secrets/kubernetes.io/serviceaccount) and 'nfs-client-root' (path: /persistentvolumes).
- Resource Usage:** CPU: 0.002, Memory: 11.908 Mi.
- Other Info:** IMAGE PULL POLICY: Always, AGE: 3d 23h.

항목	설명
이름	컨테이너 이름
상태	컨테이너 상태
재실행	컨테이너 재실행 횟수
이미지	컨테이너 이미지 이름
CPU	컨테이너 5분간 CPU 사용량
메모리	컨테이너 5분간 Memory 사용량
이미지 가져오기 정책	컨테이너 재시작 정책
AGE	컨테이너 생성 후 지난 시간
로그	컨테이너 로그 조회
터미널	컨테이너 터미널 접속
펼침 버튼	컨테이너 Arguments, Env, Volume 정보 조회

이벤트 정보

파드에서 발생하는 이벤트 정보를 제공한다.

이벤트 ^			
타입 ↑	메시지	원인	생성된 시간
Normal	Pulling image "10.20.200.200:30001/accregistry/tomtes-9acd269e-45851c1c:22"	Pulling	6m 49s
Normal	Back-off pulling image "10.20.200.200:30001/accregistry/tomtes-9acd269e-45851c1c:22"	BackOff	19d

항목	설명
타입	이벤트 타입 <ul style="list-style-type: none"> Normal: 일반적인 작업으로 발생한 이벤트 Warning: 오류에 의해 발생한 이벤트
메시지	이벤트 메시지
원인	이벤트 발생 이유
생성된 시간	이벤트 생성 후 지난 시간

컨디션 정보

컨디션 v				
타입 ↑	상태	업데이트	메시지	원인
ContainersReady	True	3d 1h	-	-
Initialized	True	3d 1h	-	-
PodScheduled	True	3d 1h	-	-
Ready	True	3d 1h	-	-

항목	설명
타입	파드 컨디션의 이름 <ul style="list-style-type: none"> PodReadyToStartContainers: 파드가 컨테이너를 시작할 준비가 되었다. PodScheduled: 파드가 노드에 스케줄되었다. ContainersReady: 파드의 모든 컨테이너가 준비되었다. Initialized: 모든 초기화 컨테이너가 성공적으로 완료(completed)되었다. Ready: 파드는 요청을 처리할 수 있으며 일치하는 모든 서비스의 로드 밸런싱 풀에 추가되어야 한다.
상태	해당 컨디션이 적용 가능한지 여부
업데이트	파드가 한 상태에서 다른 상태로 전환까지 걸린 시간
메시지	마지막 상태 전환에 대한 세부 정보

항목	설명
원인	컨디션의 마지막 전환에 대한 이유

파드 세부 정보

파드에 대한 세부 정보를 제공한다.

노드 선택터	-
호스트 IP	10.20.200.224
서비스 어카운트	alert-apiserver
DNS 정책	ClusterFirst
재시작 정책	Always
활성 마감 시간	-
서비스 품질 클래스 (QoSClass)	BestEffort

- 라벨 >
- 어노테이션 >
- 톨러레이션 >
- 어피니티 >

토폴로지

파드와 관련된 리소스를 토폴로지 형태로 보여준다.

파드의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 파드, 디플로이먼트, 레플리카셋, 스테이트풀셋, 데몬셋, 잡
- 서비스
- 서비스어카운트
- 퍼시스턴트볼륨클레임
- 시크릿, 컨피그맵
- 컨테이너, 볼륨 (해당 리소스들은 단순 연관 관계를 출력하는 리소스로 클릭하여도 화면은 이동하지 않는다.)

개요 로그 터미널 **토폴로지** YAML

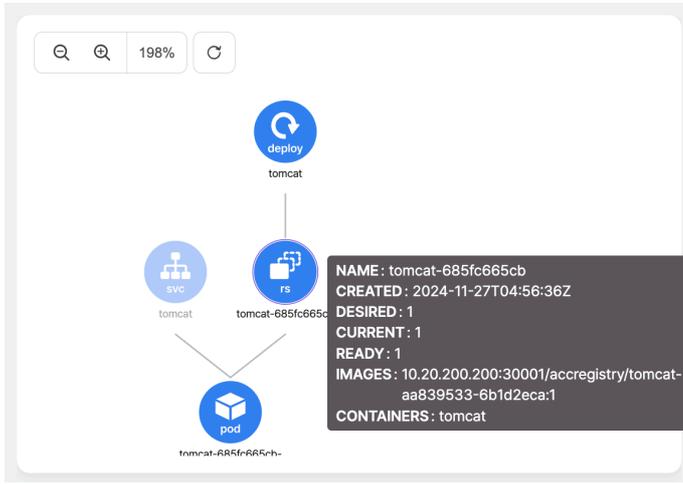
po **tomcat-685fc665cb-vh4vx** 삭제

🔍 🔍 100% 🔄

```

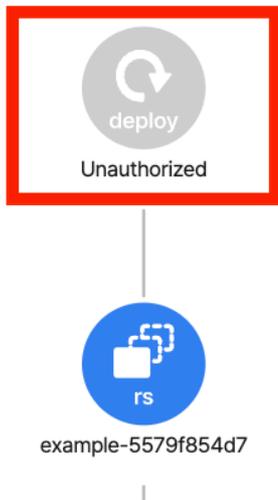
graph TD
    tomcat_deploy[tomcat] --- tomcat_svc[tomcat]
    tomcat_deploy --- tomcat_depl[tomcat-685fc665cb]
    tomcat_depl --- pod[pod]
    pod --- default[default]
    pod --- tomcat_cont[tomcat]
    tomcat_cont --- localtime[vol]
    tomcat_cont --- kube_api_access[kube-api-access-zmth]
    tomcat_cont --- scoutar_agent[vol]
  
```

리소스에 마우스 오버 시, 직접적으로 연관이 있는 리소스만 활성화되며, 리소스와 관련된 정보가 툴팁으로 보여진다. 리소스 클릭 시에는 해당 리소스의 화면으로 이동한다.



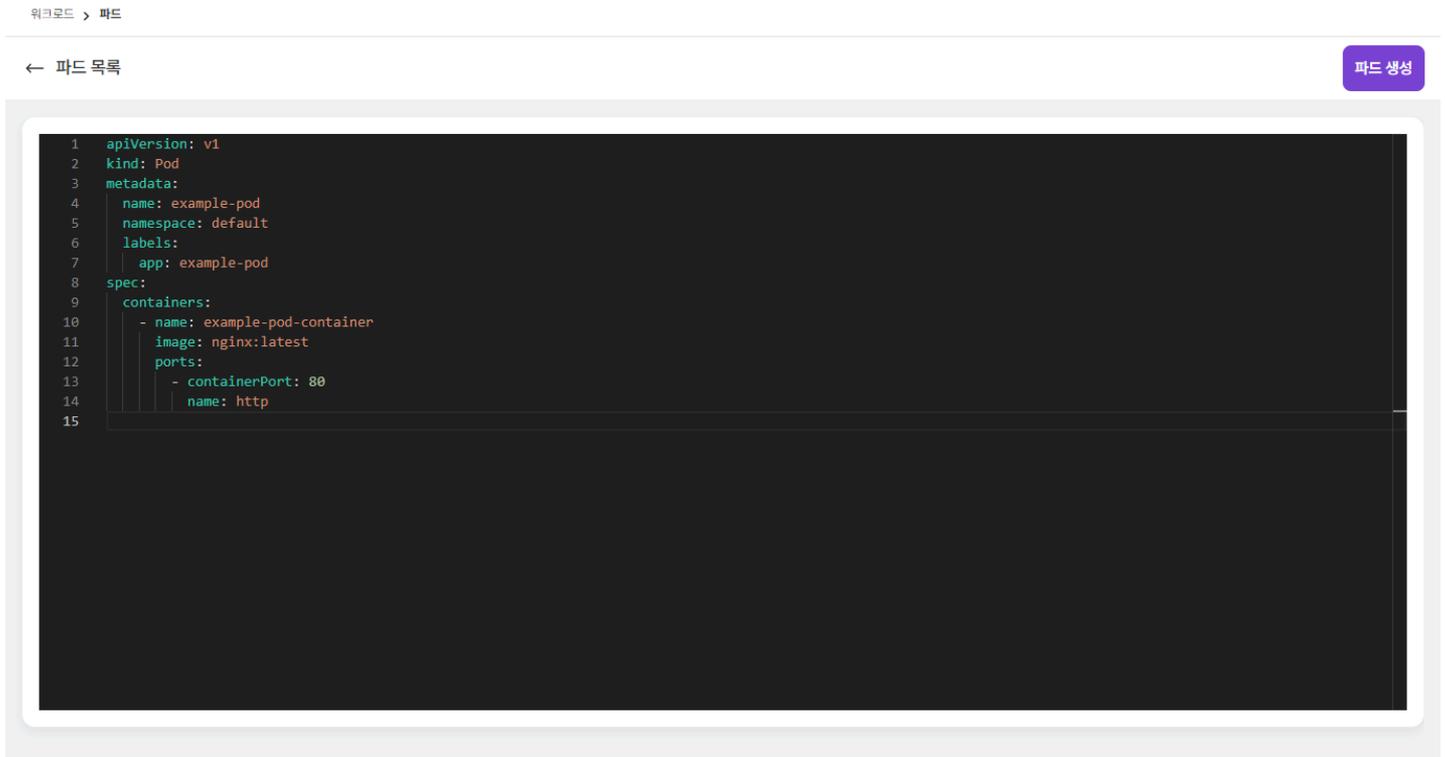
좌측 상단의 새로고침 버튼을 클릭할 경우, 토폴로지를 다시 로드한다.

만약 권한이 없는 리소스가 연관관계에 있을 경우에는, 해당 이미지와 같이 리소스의 이름 없이 비활성화된 노드 형태로 리소스의 종류만 출력하며 클릭하여도 화면 이동이 일어나지 않는다.



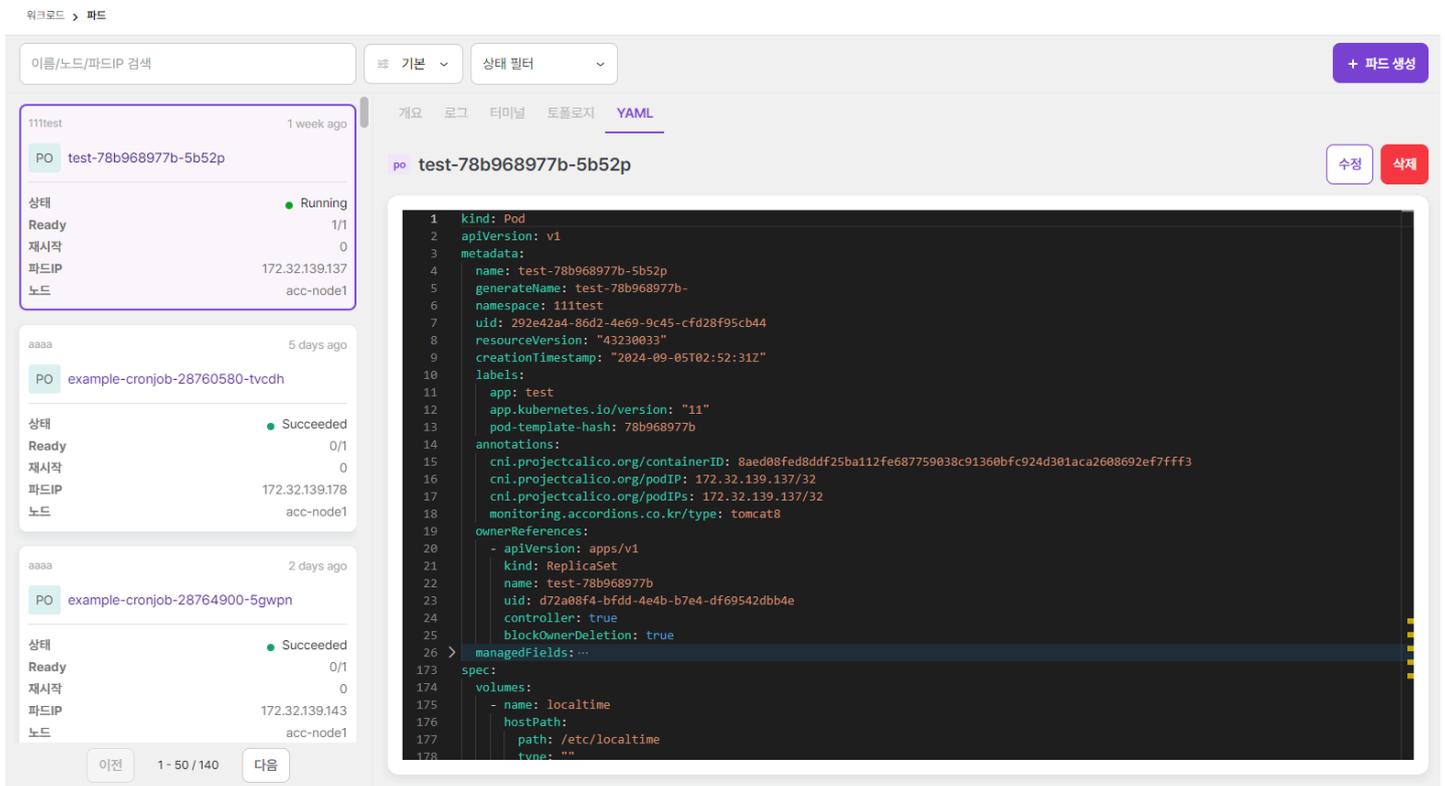
파드 생성

+ 파드 생성 을 선택하면 나타나는 화면에서 쿠버네티스 파드 리소스 정보를 입력하여 생성할 수 있다.



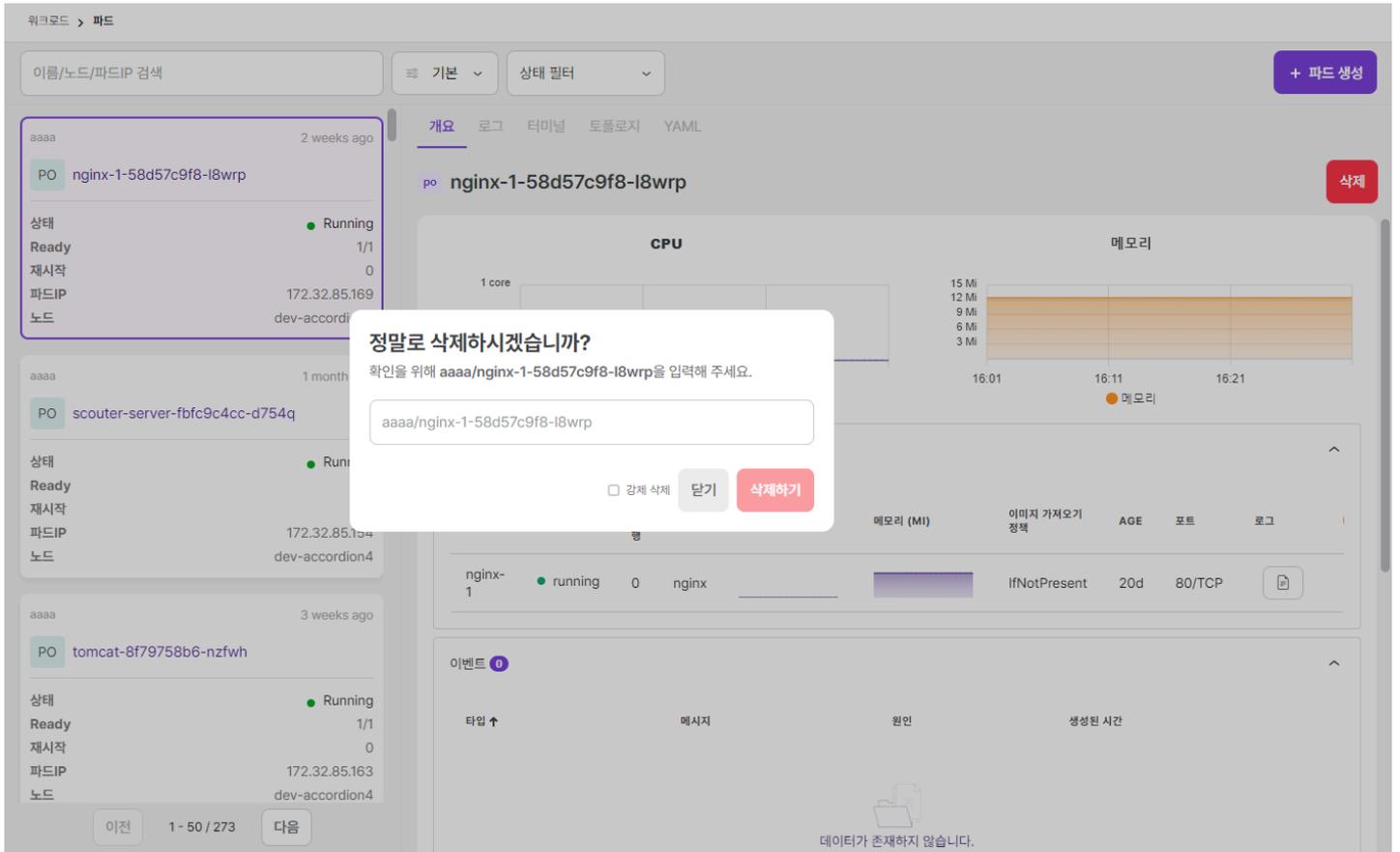
파드 수정

수정하려는 파드를 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.



파드 삭제

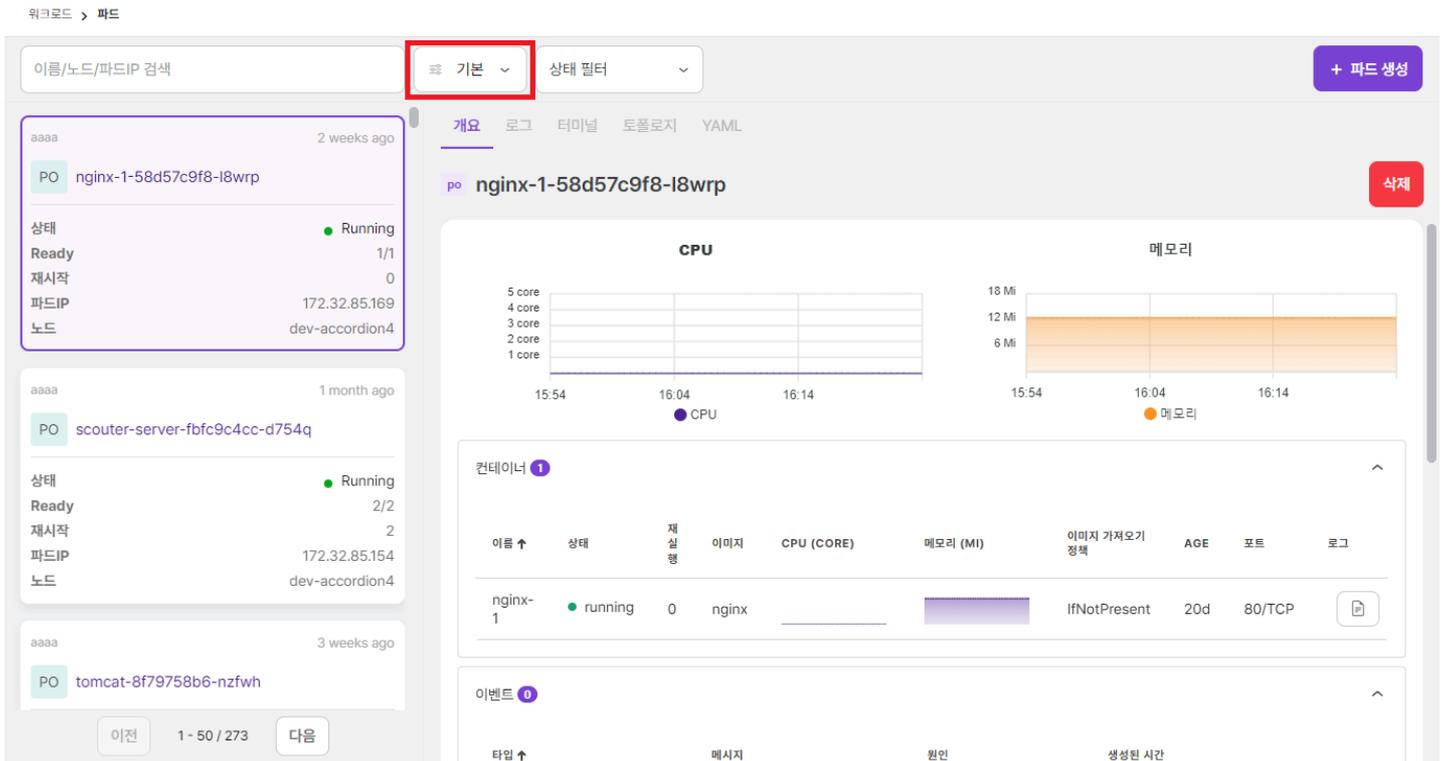
삭제하려는 파드를 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 파드 이름을 입력하여 삭제한다.

파드 정렬

파드의 정렬을 지정하여 조회한다.





이름	설명
최신순	파드의 실행된 시간이 최신인 순으로 파드를 조회한다.
오래된순	파드의 실행된 시간이 오래된순으로 파드를 조회한다.
이름순 (오름차순)	파드 이름의 오름차순으로 파드를 조회한다.
이름순 (내림차순)	파드 이름의 내림차순으로 파드를 조회한다.
네임스페이스순 (오름차순)	네임스페이스 이름의 오름차순으로 파드를 조회한다.
네임스페이스순 (내림차순)	네임스페이스 이름의 내림차순으로 파드를 조회한다.
노드 (오름차순)	파드가 속한 노드이름의 오름차순으로 파드를 조회한다.
노드 (내림차순)	파드가 속한 노드이름의 내림차순으로 파드를 조회한다.
파드IP (오름차순)	파드IP의 오름차순으로 파드를 조회한다.
파드IP (내림차순)	파드IP의 내림차순으로 파드를 조회한다.

파드 상태 필터링 검색

특정 파드 상태만을 조회할 때 사용한다.

워크로드 > 파드

이름/노드/파드IP 검색

기본 상태 필터

+ 파드 생성

개요 로그 터미널 도플로지 YAML

nginx-1-58d57c9f8-l8wrp 삭제

CPU

메모리

컨테이너 1

이름 ↑	상태	재실행	이미지	CPU (CORE)	메모리 (MI)	이미지 가져오기 정책	AGE	포트
nginx-1	● running	0	nginx	0.000	12.441	IfNotPresent	20d	80/TCP

이전 1 - 50 / 273 다음

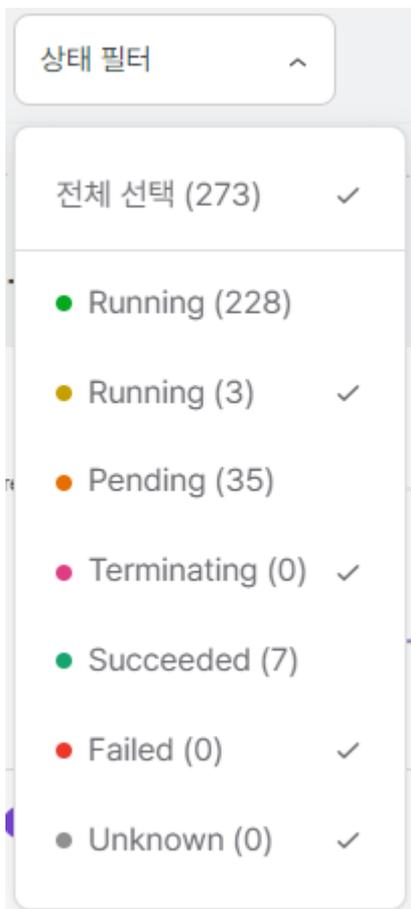
상태 필터

- 전체 선택 (273) ✓
- Running (228) ✓
- Running (3) ✓
- Pending (35) ✓
- Terminating (0) ✓
- Succeeded (7) ✓
- Failed (0) ✓
- Unknown (0) ✓

항목	설명
● Running	동작 중인 파드를 의미한다.
● Running	동작 중이지만 비정상적인 컨테이너가 존재하는 파드를 의미한다.
● Pending	실행할 준비가 되지 않은 파드를 의미한다.
● Terminating	종료 중인 파드를 의미한다.
● Succeeded	성공적으로 종료된 파드를 의미한다.
● Failed	내부 컨테이너가 실패로 종료된 파드를 의미한다.
● Unknown	파드의 상태를 얻을 수 없는 경우를 의미한다.

전체 선택 을 통해 Pod의 필터링 상태를 일괄적으로 선택하거나 해제할 수 있다.

필터링 상태는 다중 선택이 가능하다. 기본값은 전체 선택 이다.



4.2.6.3. 디플로이먼트

배포된 디플로이먼트 정보를 제공한다.

개요

배포된 디플로이먼트의 메트릭, 파드, 이벤트, 컨디션, 세부 정보를 제공한다.

The screenshot shows the 'nginx-1' deployment overview in a web interface. On the left, there is a list of other deployments: 'nginx-1' (Ready, 1/1), 'scouter-server' (Ready, 1/1), 'tomcat' (Ready, 1/1), 'alert-apiserver' (Ready, 0/0), and 'chartmuseum-chartmuseum' (Ready, 1/1). The main area displays the 'nginx-1' deployment details, including a 'CPU' graph showing 1 core usage and a '메모리' (Memory) graph showing usage around 12.441 Mi. Below the graphs is a table of pods with columns for name, status, age, replicas, node, and IP. One pod is shown in a 'Running' state. There are also sections for '이벤트' (Events) and '컨디션' (Conditions), both currently empty.

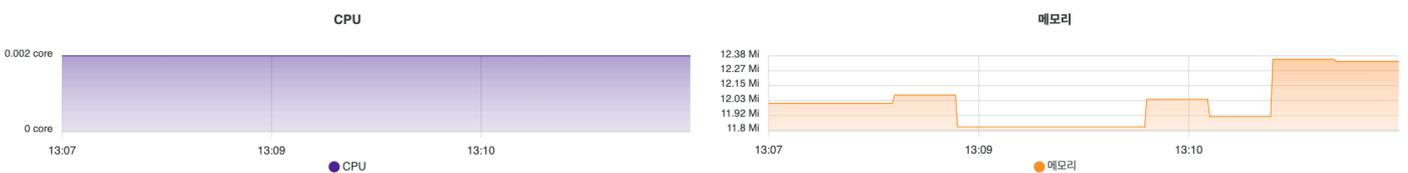
스케일 및 롤아웃

A set of control buttons for scaling and rolling out the deployment. It includes a blue button with '2/2' indicating the current number of replicas, and three white buttons with minus, plus, and refresh symbols.

항목	설명
-	디플로이먼트의 Replicas를 감소하여 파드 개수를 줄인다.
+	디플로이먼트의 Replicas를 증가하여 파드 개수를 늘린다.
롤아웃	디플로이먼트를 롤링 재시작한다.

메트릭 정보

메트릭 정보는 현재 시간으로부터 5분간 CPU, Memory 사용량을 제공한다.



파드 정보

배포된 파드에 대한 정보를 제공한다. 클릭하면 해당 파드 화면으로 이동한다.

Pod 1 ▼

이름 수	상태	AGE	재실행	노드	파드 IP	CPU (CORE)	메모리 (MI)	로그	터미널	
cicd-trigger-manager-54865f5894-f2l7c	● Running	8h 11m	0	acc-node2	172.32.24.62	<div style="width: 0.002%; height: 10px; background-color: #666;"></div> 0.002	<div style="width: 31.32%; height: 10px; background-color: #666;"></div> 31.32			

항목	설명
이름	파드 이름
상태	파드 상태
AGE	파드가 생성되고나서 지난 시간
재실행	파드 재실행 횟수
노드	파드가 배포된 노드
파드 IP	파드 고유 IP
CPU (CORE)	CPU 사용량
메모리 (MI)	MEMORY 사용량
로그	파드 로그 조회
터미널	파드 터미널 접속
삭제	파드 삭제

이벤트 정보

파드에서 발생하는 이벤트 정보를 제공한다.

이벤트 5 ▼

타입 수	메시지	원인	생성된 시간
Normal	Pulling image "registry.accordions.co.kr/console:2.4.0"	Pulling	4m 57s
Normal	Successfully pulled image "registry.accordions.co.kr/console:2.4.0" in 32.379131ms	Pulled	4m 57s
Normal	Created container webserver	Created	4m 57s
Normal	Started container webserver	Started	4m 57s
Normal	Successfully assigned acc-global/console-58f4d9dd94-gzxvn to acc-gpu1	Scheduled	4m 58s

항목	설명
타입	이벤트 타입 <ul style="list-style-type: none"> Normal: 일반적인 작업으로 발생한 이벤트 Warning: 오류에 의해 발생한 이벤트

항목	설명
메시지	이벤트 메시지
원인	이벤트 발생 이유
생성된 시간	이벤트 생성 후 지난 시간

컨디션 정보

타입 ↑	상태	업데이트	메시지	원인
Available	True	1d	Deployment has minimum availability.	MinimumReplicasAvailable
Progressing	True	1d 17h	ReplicaSet "war-6769c9bbff" has successfully progressed.	NewReplicaSetAvailable

항목	설명
타입	디플로이먼트 컨디션의 이름 <ul style="list-style-type: none"> Available: 디플로이먼트가 사용 가능하다. (기준: 디플로이먼트의 리플리카 값) Progressing: 리플리카셋에 대한 배포 진행 상황을 표시한다. ReplicaFailure: 디플로이먼트의 파드들 중 생성/삭제에 실패하는 경우 해당 컨디션이 추가된다.
상태	해당 컨디션이 적용 가능한지 여부
업데이트	디플로이먼트가 한 상태에서 다른 상태로 전환까지 걸린 시간
메시지	마지막 상태 전환에 대한 세부 정보
원인	컨디션의 마지막 전환에 대한 이유

세부 정보

디플로이먼트에 대한 세부 정보를 제공한다.

선택터	app=alert-apiserver
노드 선택터	-
소유자 참조	-
전략 유형	RollingUpdate
최대 불가	25%
최대 서지	25%
진행 기간 시간(초)	600
최소 대기 시간(초)	-

- 라벨 >
- 어노테이션 >
- 틀러레이션 >
- 어피니티 >

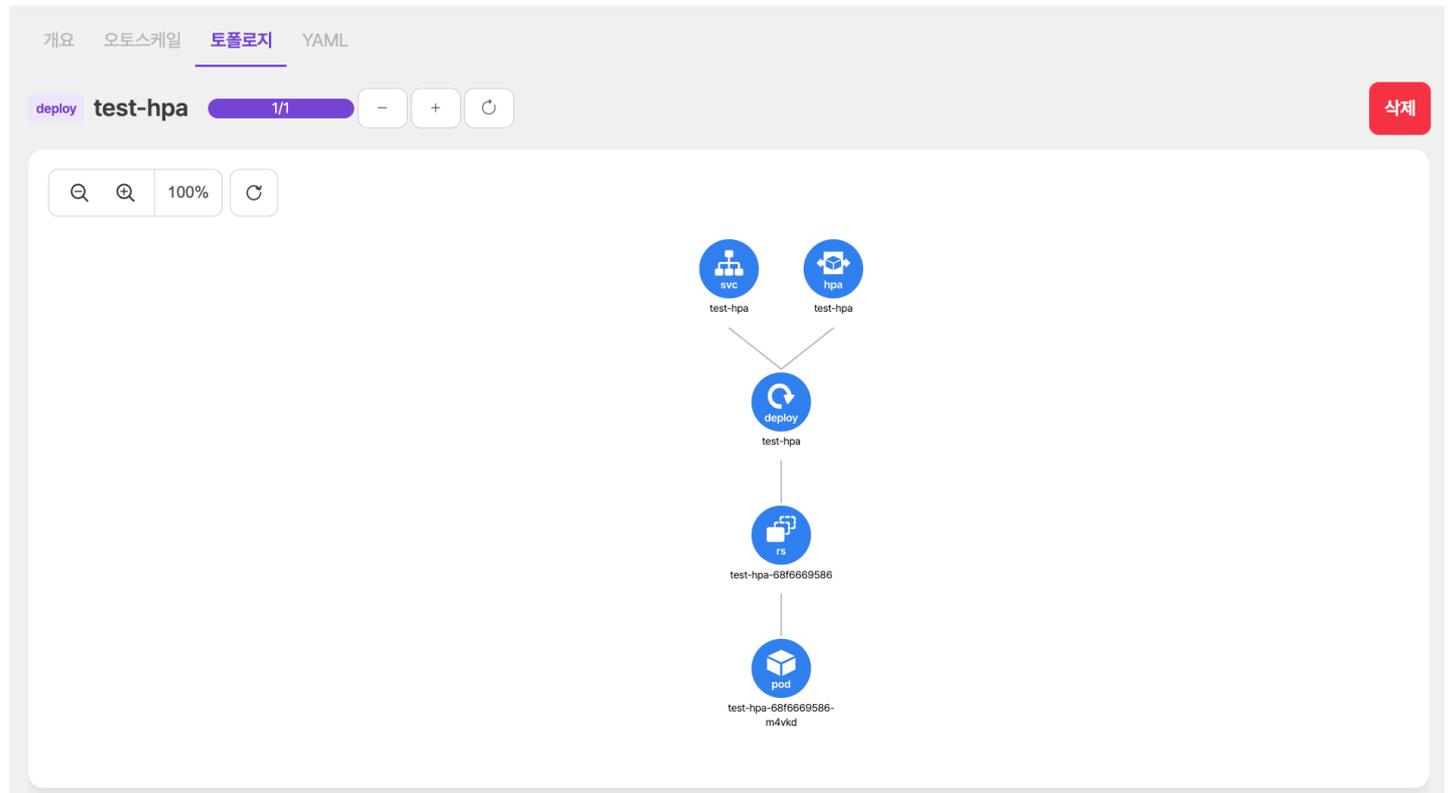
토폴로지

디플로이먼트와 관련된 리소스를 토폴로지 형태로 보여준다.

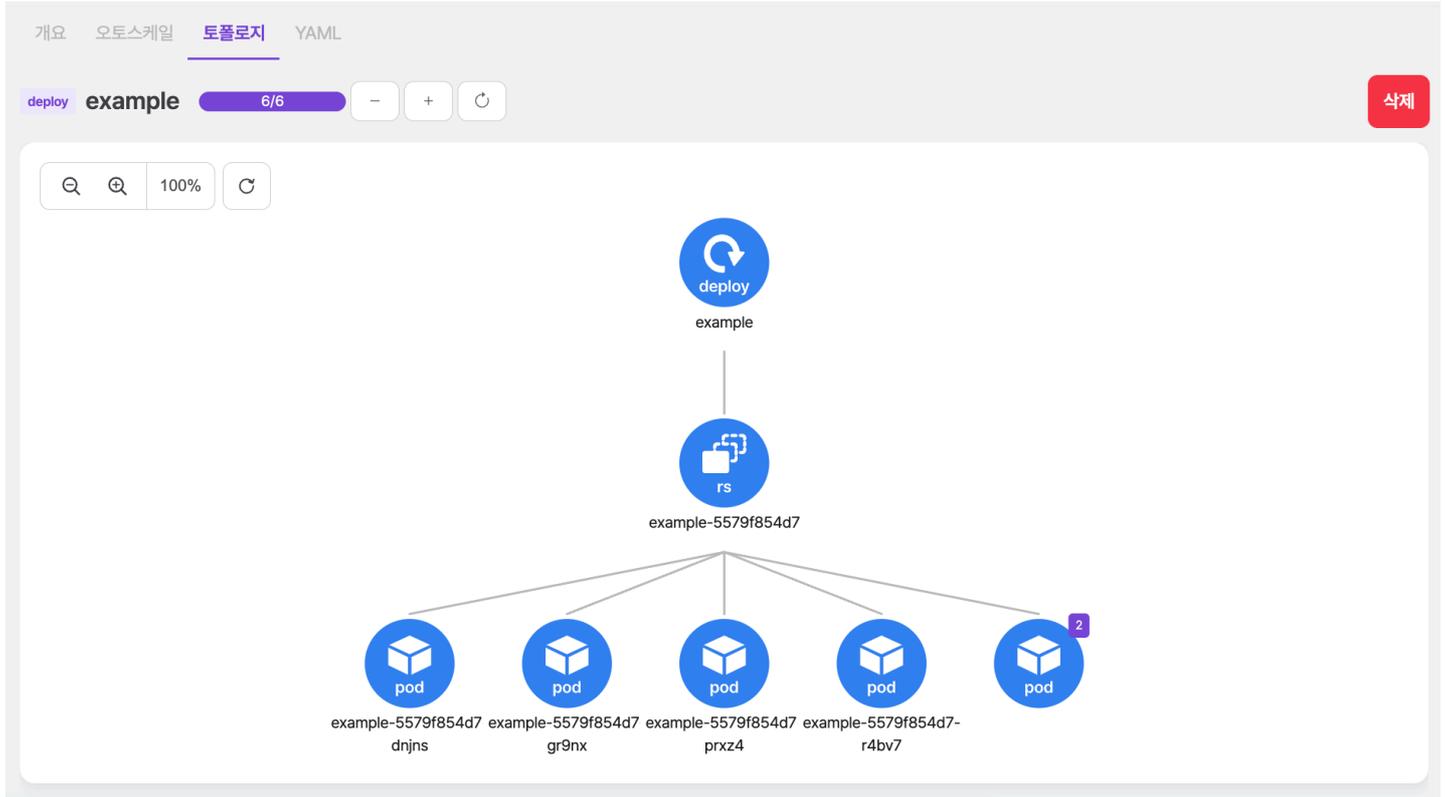
기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

디플로이먼트의 토폴로지에서 출력되는 리소스는 다음과 같다.

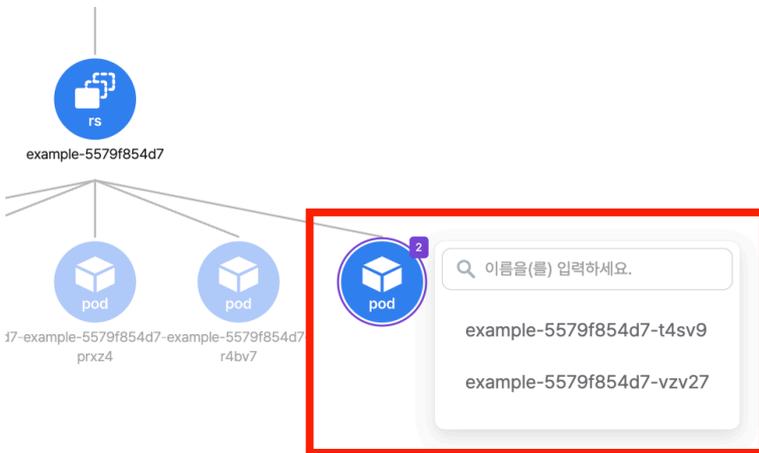
- 디플로이먼트, 레플리카셋, 파드
- HPA
- 서비스



디플로이먼트 토폴로지에서 가로로 출력되는 최대 파드 개수는 5개로 파드의 개수가 6개 이상일 경우, 다음과 같이 나머지 파드의 개수를 출력하는 노드로 보여진다.



개수를 포함하는 노드 클릭 시, 드롭다운 형태로 리스트가 나오며 파드 이름을 클릭할 경우, 해당하는 리소스 화면으로 이동한다.



디플로이먼트 생성

+ 디플로이먼트 생성 을 선택하면 나타나는 화면에서 쿠버네티스 디플로이먼트 리소스 정보를 입력하여 생성할 수 있다.

워크로드 > 디플로이먼트

← 디플로이먼트 목록

디플로이먼트 생성

```

1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: example-deployment
5    namespace: default
6    labels:
7      app: example-dp
8  spec:
9    replicas: 1
10   selector:
11     matchLabels:
12       app: example-dp
13   template:
14     metadata:
15       labels:
16         app: example-dp
17     spec:
18       containers:
19         - name: example-dp-container
20           image: nginx:latest
21           ports:
22             - containerPort: 80
23               name: http
24           volumeMounts:
25             - mountPath: /usr/share/nginx/html
26               name: example-volume
27       volumes:
28         - name: example-volume
29           emptyDir: {}
30   strategy:
31     type: RollingUpdate
32     rollingUpdate:
33       maxSurge: 25%
34       maxUnavailable: 25%
35

```

디플로이먼트 수정

수정하려는 디플로이먼트를 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

그 외에도 오토스케일 버튼으로 디플로이먼트에 대한 오토스케일 설정이 가능하다. 오토스케일은 발생 기준에 따라 메트릭 기준의 오토스케일과 시간 기준의 오토스케일로 나눌 수 있다.

TIP

아코디언에서는 디플로이먼트와 스테이트풀셋에 대해 오토스케일 기능을 제공하고 있다.

시간 기준 오토스케일	메트릭 기준 오토스케일
<p>특정 시간에 대해 파드의 개수를 스케일링</p> <ul style="list-style-type: none"> 시작 Schedule: 스케일링 시작 시간 설정 시작 Target Pods: 스케일링 시작 시 목표 파드 개수 종료 Schedule: 스케일링 종료 시간 설정 종료 Target Pods: 스케일링 종료 시 목표 파드 개수 <p>NOTE 시작 Target Pods는 종료 Target Pods보다 큰 값이어야 한다.</p>	<p>메트릭 값에 따라 파드의 개수를 스케일링</p> <ul style="list-style-type: none"> Min Pods: 최소 파드의 개수 Max Pods: 스케일링 시 최대 파드의 개수 오토스케일 기준: 스케일링 기준이 되는 메트릭 설정값: 스케일링이 발생하는 임계치 <p>NOTE Max Pods는 Min Pods보다 큰 값이어야 한다.</p>

시간 기준 오토스케일 설정 시 아래와 같이 크론 스케줄 포맷에 의해 작성한다.

시간	허용 가능 값	허용 가능 특수문자
분	0-59	* / , -
시	0-23	* / , -
일자	1-13	* / , - ?
달	1-12 or JAN-DEC	* / , -
요일	0-6 or SUN-SAT	* / , - ?

시간 기준 오토스케일 설정의 예는 아래와 같다.

```

시작 Schedule : 1 9 1 11 *
시작 Target Pods : 3
종료 Schedule : 1 9 1 12 *
종료 Target Pods : 1
    
```

= 매년 11월 1일 오전 9시 1분에 파드가 3개로 늘어났다가 매년 12월 1일 오전 9시 1분에 파드가 1개로 줄어든다.

메트릭 기준 오토스케일의 경우 기본적으로 CPU와 메모리 사용량에 따라 적용 가능하다. 아코디언에서는 그 외에 스카우터에서 수집한 메트릭에 의한 스케일링도 가능하며 해당 메트릭은 다음과 같다.

- apm_ActiveService
- apm_TPS
- apm_GcTime
- apm_ElapsedTime

메트릭 기준의 오토스케일을 설정할 때는 파드의 리소스 설정 및 스카우터 모니터링 설정에 따라 오토스케일 설정 가능 여부 및 사용할 수 있는 메트릭의 종류가 다르다. 그 기준은 아래와 같다.

조건	사용 가능한 오토스케일 기준
파드리소스 O / APM 모니터링 O	CPU, Memory + apm_ActiveService, apm_TPS, apm_GcTime, apm_ElapsedTime
파드리소스 O / APM 모니터링 X	CPU, Memory
파드리소스 X / APM 모니터링 O	apm_ActiveService, apm_TPS, apm_GcTime, apm_ElapsedTime
파드리소스 X / APM 모니터링 X	메트릭 기준의 오토스케일 사용 불가

디플로이먼트 삭제

삭제하려는 디플로이먼트를 선택하고 우측의 삭제 버튼을 선택한다.

The screenshot displays the Kubernetes dashboard interface. A modal dialog box is centered on the screen, asking for confirmation to delete a deployment. The dialog text is as follows:

정말로 삭제하시겠습니까?
 확인을 위해 `aaaa/nginx-1`을 입력해 주세요.

Below the text is an input field containing `aaaa/nginx-1`. To the right of the input field are two buttons: '닫기' (Close) and '삭제하기' (Delete).

The background interface shows a list of deployments on the left, including 'nginx-1', 'scouter-server', 'tomcat', and 'alert-apiserver'. The main area shows the details for the 'nginx-1' deployment, including its status (Running), age (20d), and resource usage (CPU: 0.000, Memory: 12.441 Mi).

모달에서 네임스페이스와 디플로이먼트 이름을 입력하여 삭제한다.

4.2.6.4. 스테이트풀셋

배포된 스테이트풀셋 정보를 제공한다.

개요

배포된 스테이트풀셋의 메트릭, 파드, 이벤트, 상태, 세부 정보를 제공한다.

스케일 및 롤아웃

항목	설명
-	스테이트풀셋의 Replicas를 감소하여 파드 개수를 줄인다.
+	스테이트풀셋의 Replicas를 증가하여 파드 개수를 늘린다.
롤아웃	스테이트풀셋을 롤링 재시작한다.

메트릭 정보

메트릭 정보는 현재 시간으로부터 5분간 CPU, Memory 사용량을 제공한다.



파드 정보

디플로이먼트와 내용이 동일하기 때문에 디플로이먼트 파드 정보를 참고한다.

이벤트 정보

파드와 내용이 동일하기 때문에 파드 이벤트 정보를 참고한다.

세부 정보

스테이트풀셋에 대한 세부 정보를 제공한다.

선택터	app=redis, release=redis, role=master
노드 선택터	-
OwnerReference	-
Pod Management Policy	OrderedReady
업데이트 전략	RollingUpdate

- 라벨 >
- 어노테이션 >
- 롤러레이션 >
- 어피니티 >
- 블룸 클레임 템플릿 >

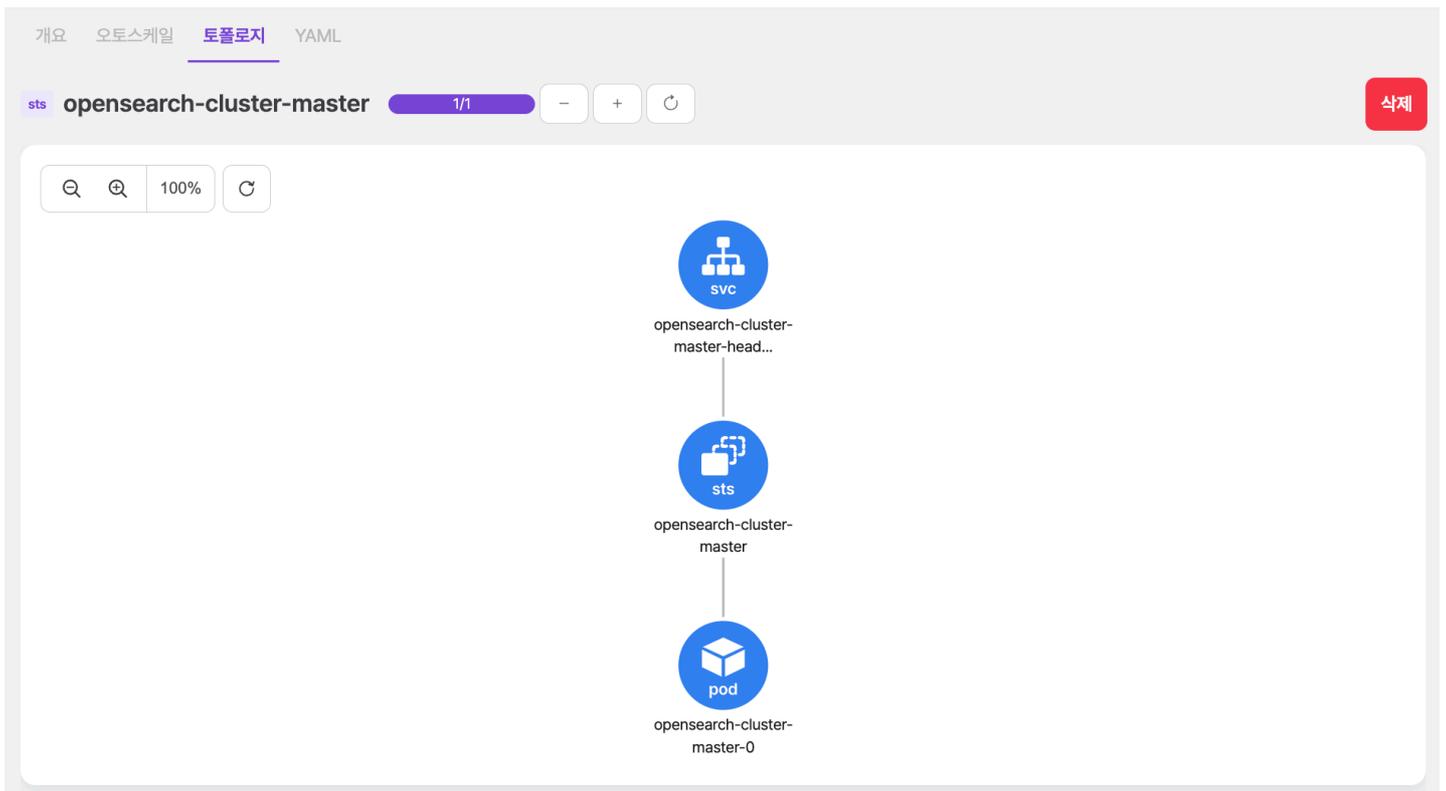
토폴로지

스테이트풀셋과 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

스테이트풀셋의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 스테이트풀셋, 파드
- 서비스



스테이트풀셋의 토폴로지에서 가로로 출력되는 최대 파드 개수는 5개로 기능은 디플로이먼트와 동일하기 때문에 디플로이먼트 토폴로지 설명을 참고한다.

스태이트풀셋 생성

+ 스테이트풀셋 생성 을 선택하면 나타나는 화면에서 쿠버네티스 스테이트풀셋 리소스 정보를 입력하여 생성할 수 있다.

클러스터

host-cluster-200

네임스페이스

전체 네임스페이스

admin

워크로드 > 스테이트풀셋

← 스테이트풀셋 목록

스태이트풀셋 생성

```

1  apiVersion: apps/v1
2  kind: StatefulSet
3  metadata:
4    name: example-statefulset
5    namespace: default
6    labels:
7      app: example-sts
8  spec:
9    serviceName: example-svc
10   replicas: 1
11   selector:
12     matchLabels:
13       app: example-sts
14   template:
15     metadata:
16       labels:
17         app: example-sts
18     spec:
19       containers:
20         - name: example-sts-container
21           image: nginx:latest
22           ports:
23             - containerPort: 80
24               name: http
25           volumeMounts:
26             - mountPath: /data
27               name: example-data
28   volumeClaimTemplates:
29     - metadata:
30       name: example-data
31     spec:
32       accessModes:
33         - ReadWriteOnce
34       storageClassName: example-storageclass
35       resources:
36         requests:
37           storage: 1Gi
38

```

스태이트풀셋 수정

수정하려는 스테이트풀셋을 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

스태이트풀셋도 디플로이먼트와 동일하게 오토스케일 설정이 가능하다.

TIP

디플로이먼트 오토스케일 가이드 참고

스테이트풀셋 삭제

삭제하려는 스테이트풀셋을 선택하고 우측의 삭제 버튼을 선택한다.

The screenshot shows the Kubernetes dashboard interface. A confirmation modal is displayed in the center, asking for confirmation to delete the StatefulSet. The modal text is as follows:

정말로 삭제하시겠습니까?
 확인을 위해 acc-system/alertmanager-main을 입력해 주세요.

The input field contains: acc-system/alertmanager-main

Buttons: 닫기 (Close), 삭제하기 (Delete)

The background interface shows the details for the StatefulSet 'alertmanager-main' in the 'acc-system' namespace. It includes a table of pods and a memory usage graph.

이름 ↑	상태	AGE	재실행	노드	파드 IP	CPU (CORE)	메모리 (MI)
alertmanager-main-0	Running	97d	2	dev-accordion5	172.32.64.137	0.001	

모달에서 네임스페이스와 스테이트풀셋 이름을 입력하여 삭제한다.

4.2.6.5. 데몬셋

배포된 데몬셋 정보를 제공한다.

개요

배포된 데몬셋의 메트릭, 파드, 이벤트, 상태, 세부 정보를 제공한다.

위크로드 > 데몬셋

이름 검색

- acc-global 10 months ago
DS gateway-proxy
Ready 3/3
- acc-system 10 months ago
DS filebeat-filebeat
Ready 5/5
- acc-system 8 months ago
DS ingress-nginx-controller
Ready 2/2
- acc-system 6 days ago
DS nginx-daemon-set
Ready 2/2
- acc-system 8 months ago
DS node-exporter
Ready 5/5
- depart 7 months ago
DS fluentd-ds
Ready 5/5

기본

ds gateway-proxy 3/3

CPU

메모리

이름 ↑	상태	AGE	재실행	노드	파드 IP	CPU (CORE)	메모리 (MI)	로그
gateway-proxy-69kqv	Running	96d	2	dev-accordion3	172.32.37.145	0.007	11.094	[Log] [Trash]
gateway-proxy-hs58d	Running	96d	2	dev-accordion2	172.32.190.62	0.002	10.195	[Log] [Trash]
gateway-proxy-klbv2	Running	96d	1	dev-accordion1	172.32.149.233	0.002	11.176	[Log] [Trash]

이벤트 0

타입 ↑	메시지	원인	생성된 시간
데이터가 존재하지 않습니다.			

롤아웃

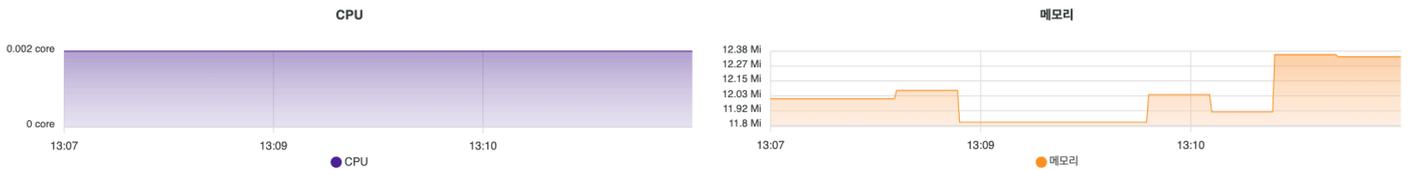
3/3

↻

항목	설명
롤아웃	데몬셋을 롤링 재시작한다.

메트릭 정보

메트릭 정보는 현재 시간으로부터 5분간 CPU, Memory 사용량을 제공한다.



파드 정보

디플로이먼트와 내용이 동일하기 때문에 디플로이먼트 파드 정보를 참고한다.

이벤트 정보

파드와 내용이 동일하기 때문에 파드 이벤트 정보를 참고한다.

세부 정보

데몬셋에 대한 세부 정보를 제공한다.

셀렉터	app=filebeat-filebeat, release=filebeat
노드 셀렉터	-
Age	90d
OwnerReference	-
업데이트 전략	RollingUpdate

- 라벨 >
- 어노테이션 >
- 톨러레이션 >
- 어피니티 >

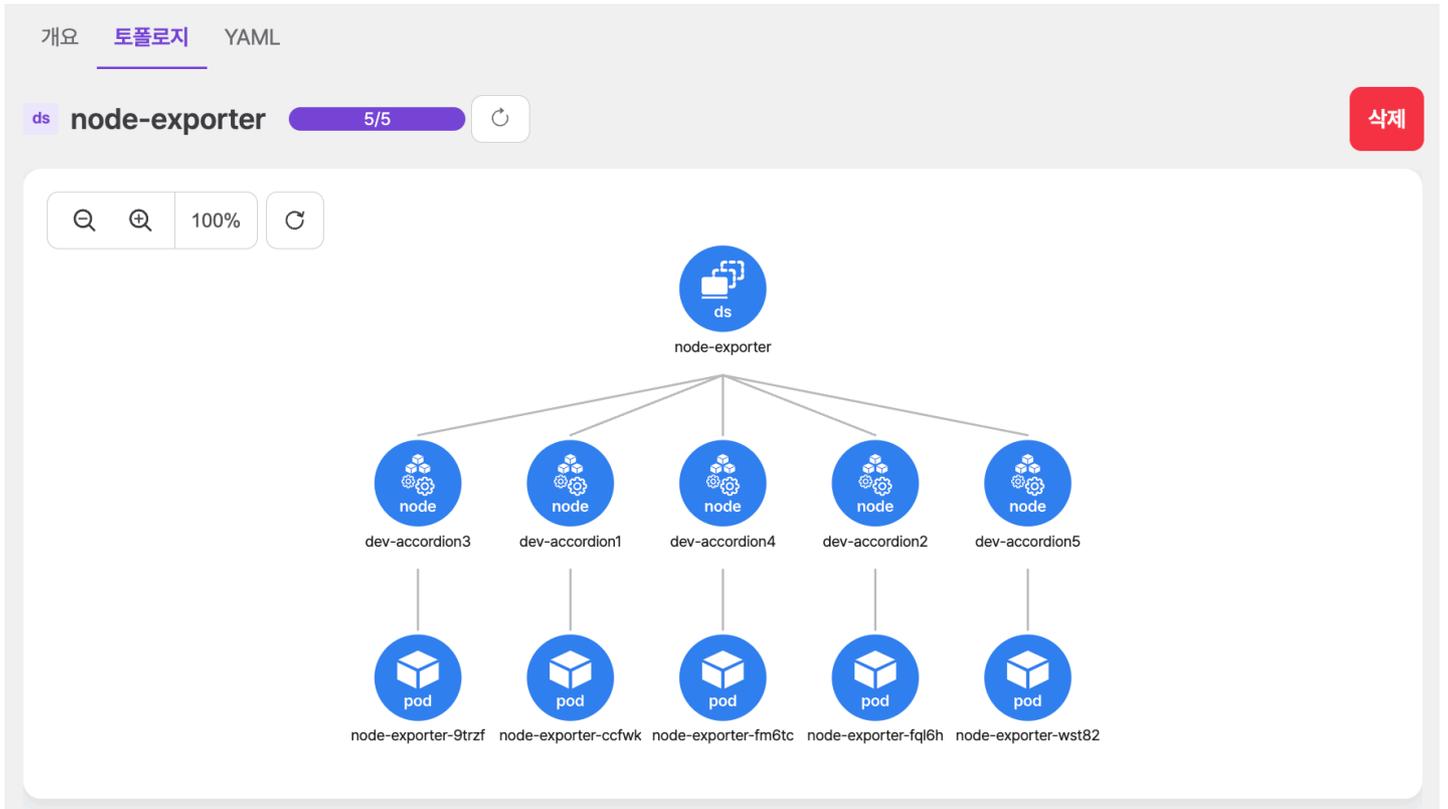
토폴로지

데몬셋과 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

데몬셋의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 데몬셋, 노드, 파드



데몬셋 생성

+ 데몬셋 생성 을 선택하면 나타나는 화면에서 쿠버네티스 데몬셋 리소스 정보를 입력하여 생성할 수 있다.

워크로드 > 데몬셋

← 데몬셋 목록

데몬셋 생성

```

1  apiVersion: apps/v1
2  kind: DaemonSet
3  metadata:
4    name: example-daemonset
5    namespace: default
6    labels:
7      app: example-ds
8  spec:
9    selector:
10   matchLabels:
11     app: example-ds
12   template:
13     metadata:
14       labels:
15         app: example-ds
16     spec:
17       containers:
18         - name: example-ds-container
19           image: nginx:latest
20           ports:
21             - containerPort: 80
22               name: http
23           volumeMounts:
24             - mountPath: /usr/share/nginx/html
25               name: example-volume
26           volumes:
27             - name: example-volume
28               emptyDir: {}
29

```

데몬셋 수정

수정하려는 데몬셋을 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

데몬셋 삭제

삭제하려는 데몬셋을 선택하고 우측의 삭제 버튼을 선택한다.

The screenshot shows the Kubernetes dashboard interface. On the left, a list of DaemonSets is visible, including 'gateway-proxy' under the 'acc-global' namespace. The main area displays the 'gateway-proxy' DaemonSet details, showing its status as '3/3' and a '삭제' (Delete) button. A modal dialog is overlaid in the center, asking '정말로 삭제하시겠습니까?' (Are you sure you want to delete?) and providing a confirmation field with the text 'acc-global/gateway-proxy'. Below the dialog, there are '닫기' (Close) and '삭제하기' (Delete) buttons. The background shows performance graphs for CPU and memory usage.

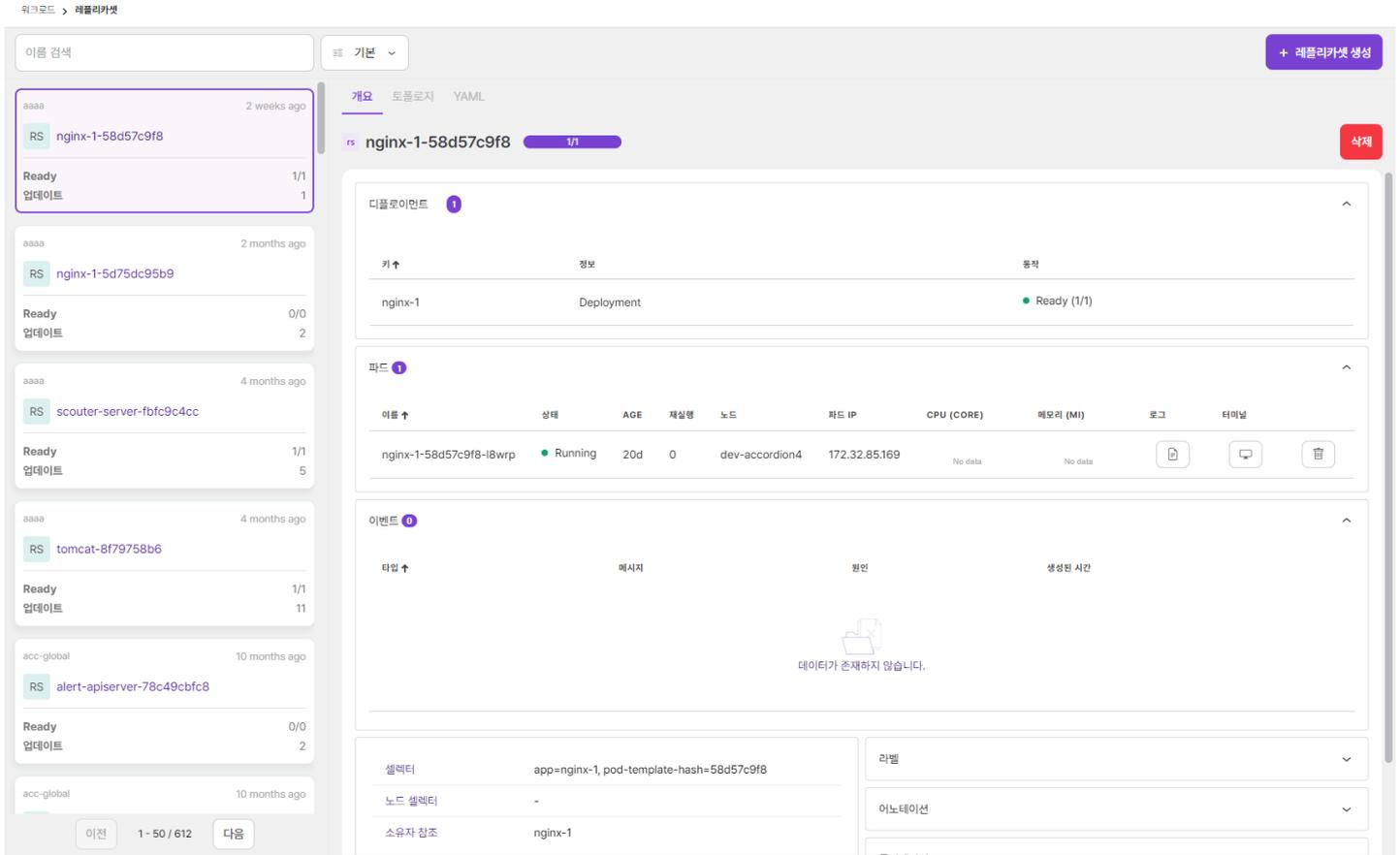
모달에서 네임스페이스와 데몬셋 이름을 입력하여 삭제한다.

4.2.6.6. 레플리카셋

배포된 레플리카셋 정보를 제공한다.

개요

배포된 레플리카셋의 디플로이먼트, 파드, 이벤트, 세부 정보를 제공한다.



디플로이먼트 정보

레플리카셋의 ownerReference인 디플로이먼트 정보를 제공하며 클릭하면 해당 디플로이먼트 화면으로 이동한다.

파드 정보

디플로이먼트와 내용이 동일하기 때문에 디플로이먼트 파드 정보를 참고한다.

이벤트 정보

파드와 내용이 동일하기 때문에 파드 이벤트 정보를 참고한다.

세부 정보

레플리카셋에 대한 세부 정보를 제공한다.

선택기	app=alert-apiserver, pod-template-hash=5b78464858
노드 선택기	-
소유자 참조	alert-apiserver

라벨	>
어노테이션	>
틀러레이션	>
어피니티	>

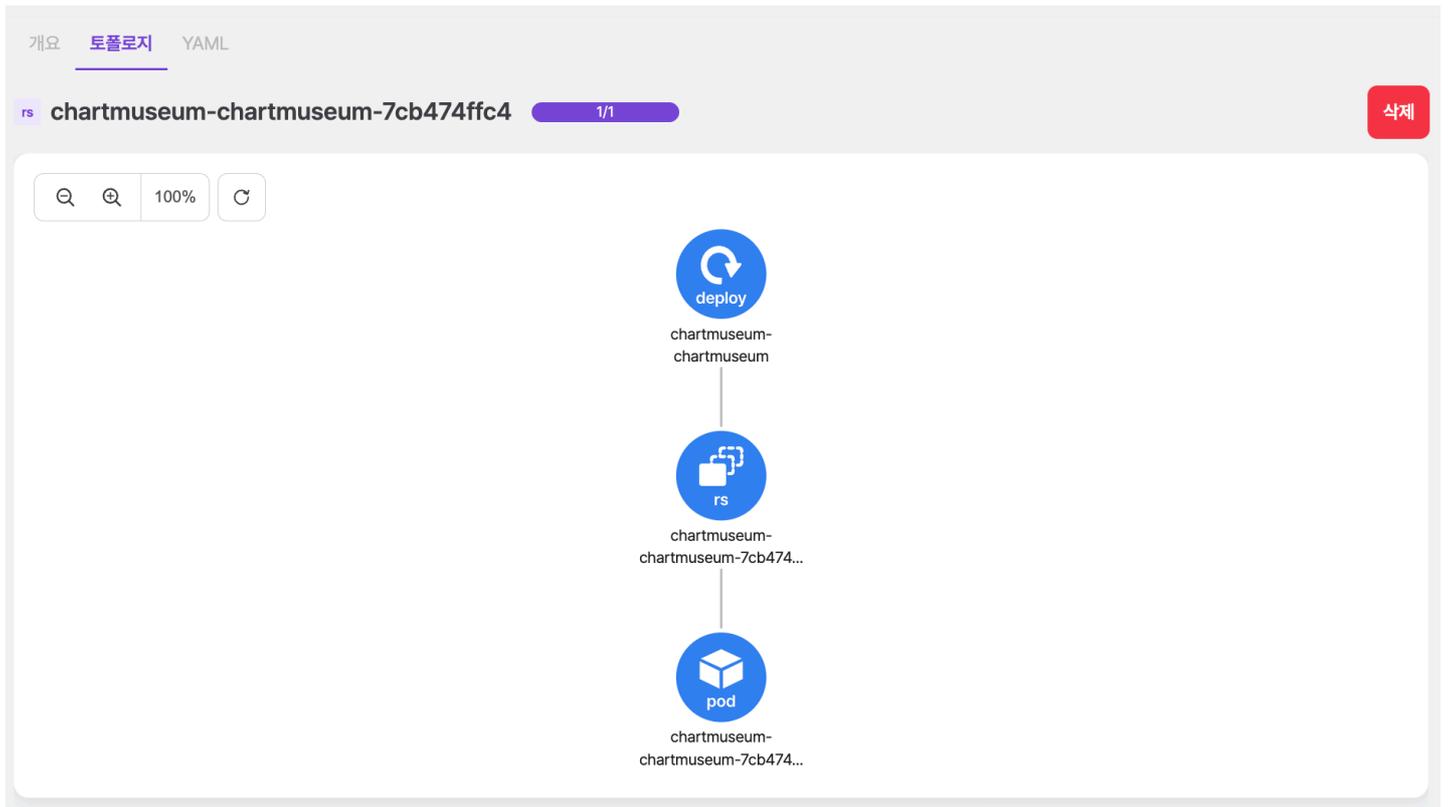
토폴로지

레플리카셋과 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

레플리카셋의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 디플로이먼트, 레플리카셋, 파드



레플리카셋의 토폴로지에서 가로로 출력되는 최대 파드 개수는 5개로 기능은 디플로이먼트와 동일하기 때문에 디플로이먼트 토폴로지 설명을 참고한다.

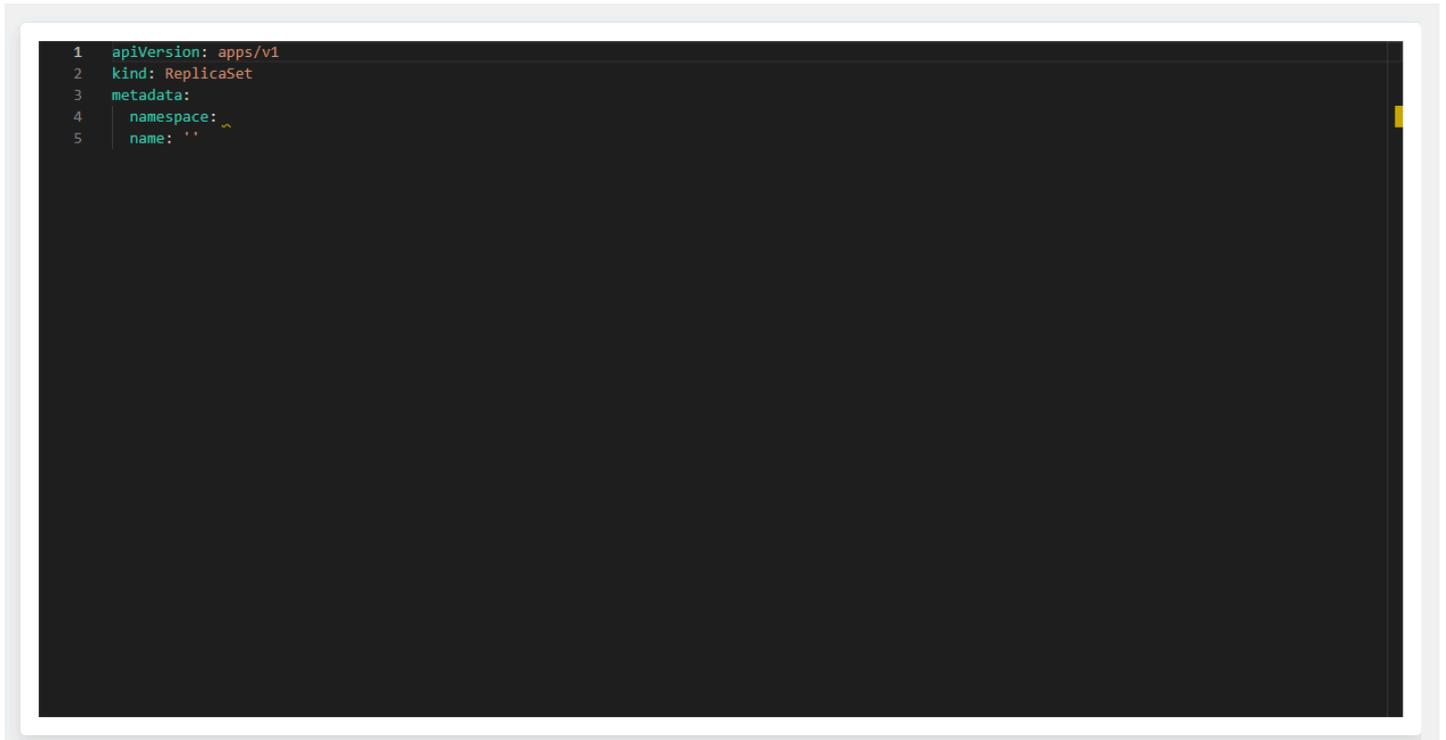
레플리카셋 생성

+ 레플리카셋 생성 을 선택하면 나타나는 화면에서 쿠버네티스 레플리카셋 리소스 정보를 입력하여 생성할 수 있다.

워크로드 > 레플리카셋

← 레플리카셋 목록

레플리카셋 생성

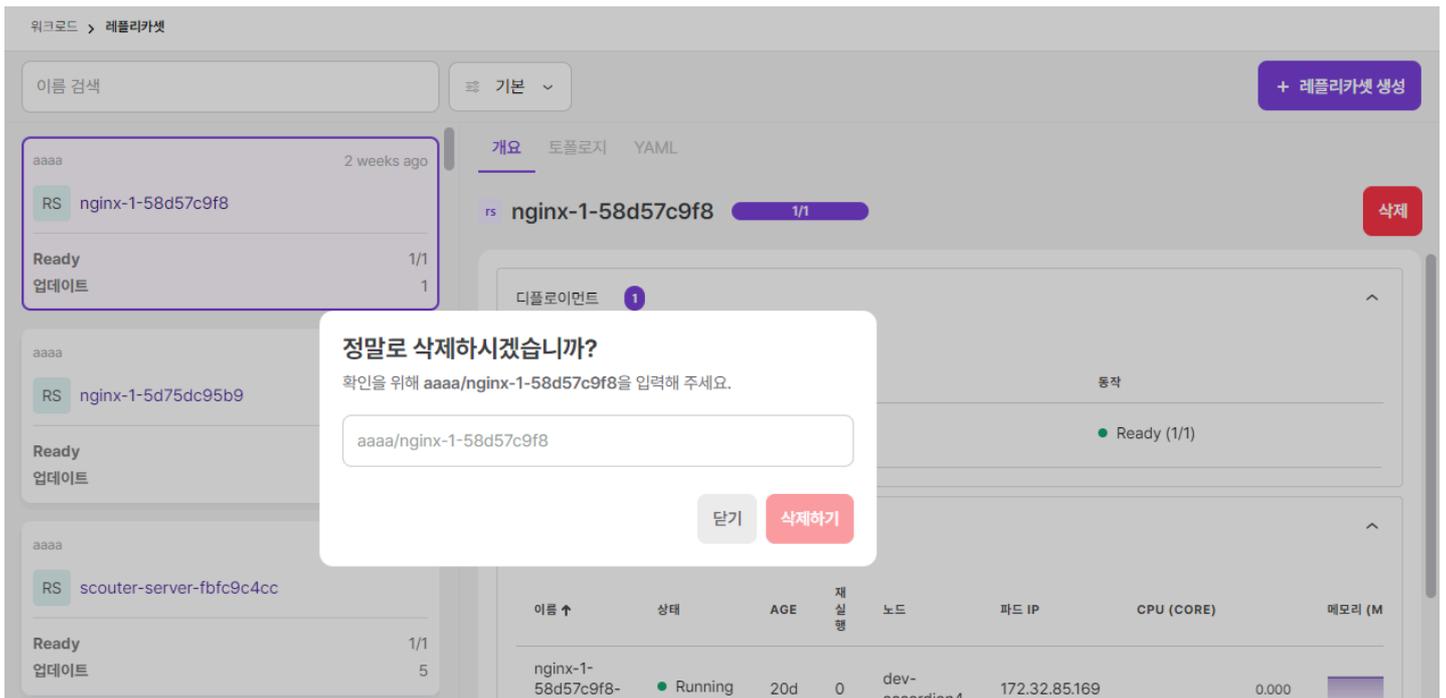


레플리카셋 수정

수정하려는 레플리카셋을 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

레플리카셋 삭제

삭제하려는 레플리카셋을 선택하고 우측의 삭제 버튼을 선택한다.



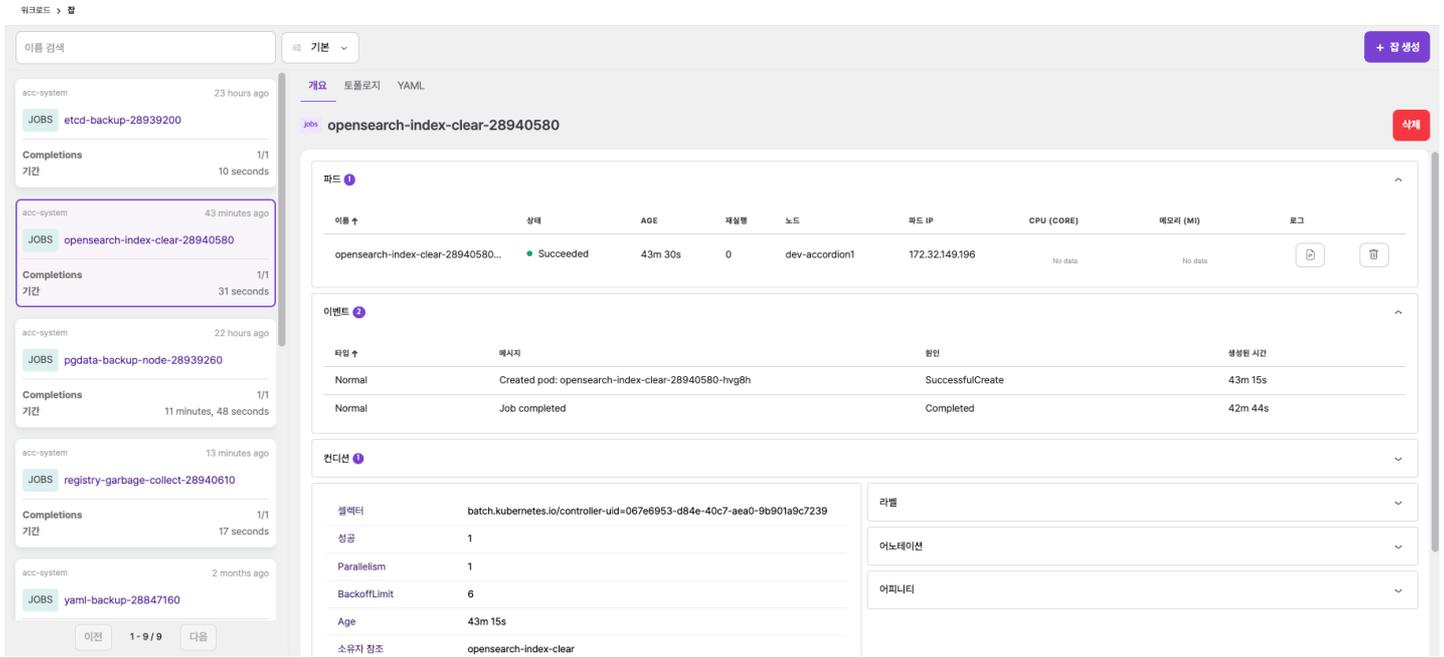
모달에서 네임스페이스와 레플리카셋 이름을 입력하여 삭제한다.

4.2.6.7. 잡

배포된 잡 정보를 제공한다.

개요

배포된 잡의 파드, 이벤트, 상태, 세부 정보를 제공한다.



파드 정보

디플로이먼트와 내용이 동일하기 때문에 디플로이먼트 파드 정보를 참고한다.

이벤트 정보

파드와 내용이 동일하기 때문에 파드 이벤트 정보를 참고한다.

세부 정보

잡에 대한 세부 정보를 제공한다.

셀렉터	batch.kubernetes.io/controller-uid=067e6953-d84e-40c7-aea0-9b901a9c7239
성공	1
Parallelism	1
BackoffLimit	6
Age	43m 15s
소유자 참조	opensearch-index-clear
시작 시간	2025-01-10 00:00:00
완료 시간	2025-01-10 00:00:31
활성 마감 시간	-

라벨	▼
어노테이션	▼
어피니티	▼

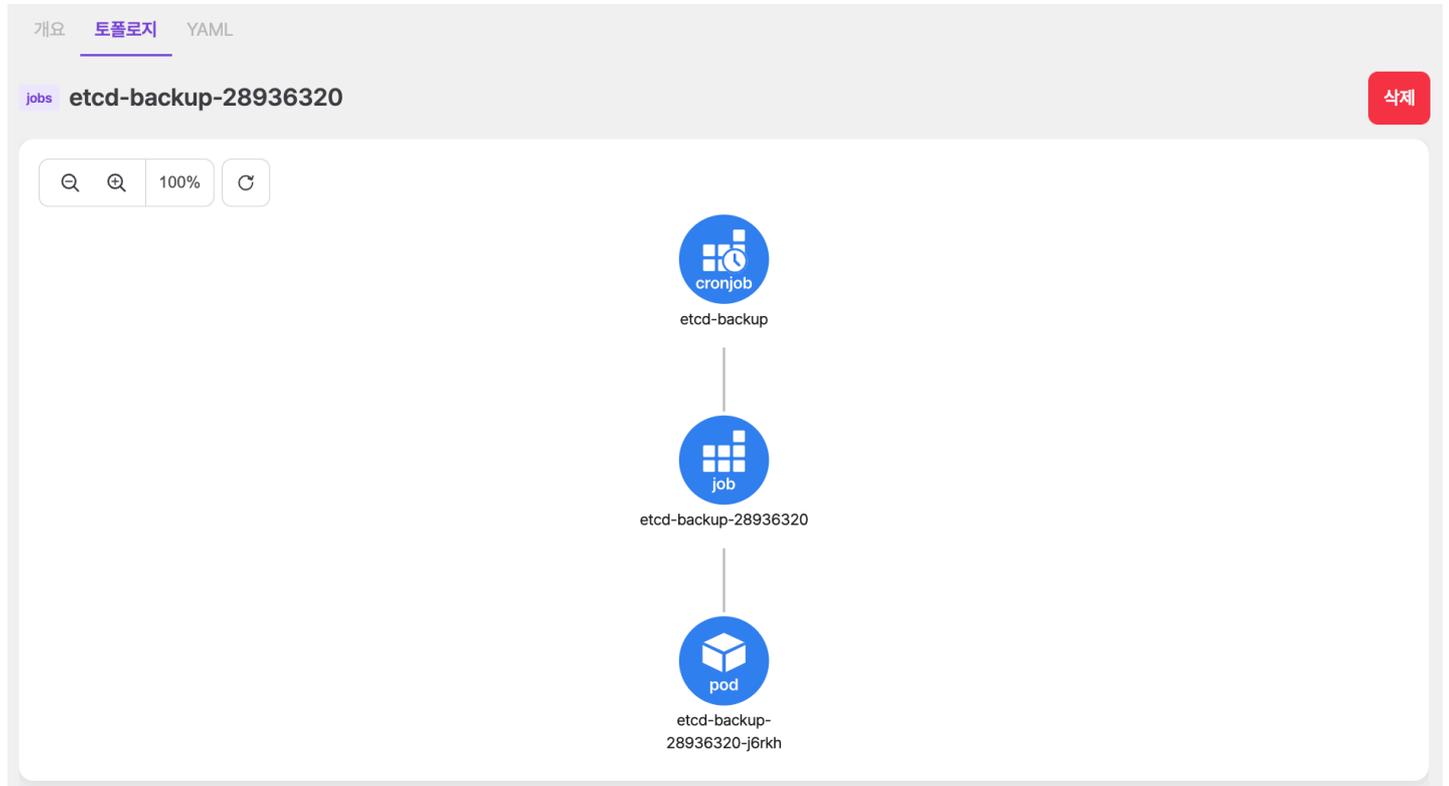
토폴로지

잡과 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

잡의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 크론잡, 잡, 파드



잡 생성

+ 잡 생성 을 선택하면 나타나는 화면에서 쿠버네티스 잡 리소스 정보를 입력하여 생성할 수 있다. 생성 시에는 FORM/YAML로 입력할 수 있다.

← 잡 목록 FORM YAML 잡 생성

이름 *

네임스페이스 *

컨테이너명 *

컨테이너 이미지 *

명령어

인수

환경 변수

키	값
<input type="button" value="추가"/>	

포트

포트명	포트	프로토콜
<input type="button" value="추가"/>		

리소스

CPU 요청	CPU 제한
<input type="text" value="CPU 요청을 입력하세요."/>	<input type="text" value="CPU 제한을 입력하세요."/>
메모리 요청	메모리 제한
<input type="text" value="메모리 요청을 입력하세요."/>	<input type="text" value="메모리 제한을 입력하세요."/>

Completions

Parallelsim

항목	설명
이름	잡 이름
네임스페이스	잡이 생성될 네임스페이스
컨테이너명	잡에서 실행될 컨테이너명
컨테이너 이미지	잡에서 실행될 컨테이너 이미지
명령어	컨테이너가 시작될 때 실행될 명령어
인수	명령어에 전달될 인수
환경 변수가	컨테이너의 환경 변수
포트	컨테이너가 사용할 네트워크 포트

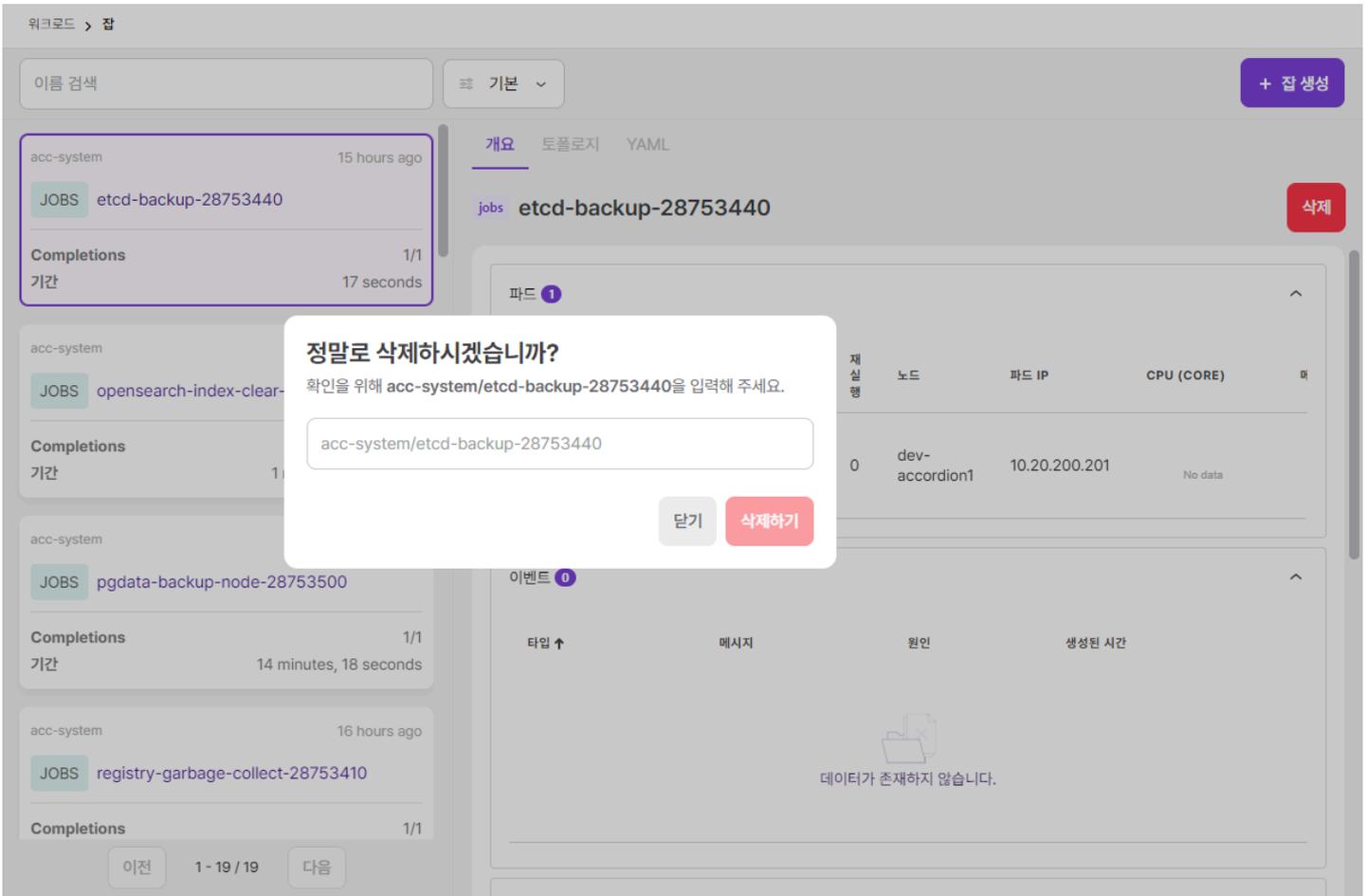
항목	설명
리소스	컨테이너에 할당될 cpu,memory 리소스 <ul style="list-style-type: none"> 요청(Requests) : 최소 필요량 제한(Limits) : 최대 허용량을 지정
완료	잡이 성공적으로 완료되어야 하는 파드 개수
병렬 처리	동시에 실행될 파드 인스턴스의 수
재시작 정책	잡이 실패할 경우 재시작 여부와 정책 설정 <ul style="list-style-type: none"> OnFailure : 실패 시 재시작 Never : 실패 시 재시작하지 않음
백오프 한계	잡의 파드의 재시도 횟수 제한, 초과시 잡 실패

잡 수정

수정하려는 잡을 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

잡 삭제

삭제하려는 잡을 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 잡 이름을 입력하여 삭제한다.

4.2.6.8. 크론잡

배포된 크론잡 정보를 제공한다.

개요

배포된 크론잡의 잡, 이벤트, 세부정보를 제공한다.

잡 정보

배포된 잡에 대한 정보를 제공한다. 클릭하면 해당 잡 화면으로 이동한다.

이름 ↑	네임스페이스	COMPLETIONS	소요 시간	AGE
etcd-backup-28753440	acc-system	1/1	17s	15h 55m

항목	설명
이름	잡 이름
네임스페이스	잡이 생성된 네임스페이스
COMPLETIONS	크론잡으로 생성된 잡 완료 개수
소요 시간	잡이 실행되고 완료되기까지 시간
AGE	잡이 생성되고나서 지난 시간

항목	설명
삭제	잡 삭제

이벤트 정보

파드와 내용이 동일하기 때문에 파드 이벤트 정보를 참고한다.

세부 정보

크론잡에 대한 세부 정보를 제공한다.

Completions	-
Parallelism	-
Successful Jobs History Limit	0
Failed Jobs History Limit	1
OwnerReference	-
Concurrency Policy	Allow
LastSchedule Time	-
Starting Deadline Seconds	-
ActiveDeadline Seconds	-
Suspend	false

라벨 >

어노테이션 >

토폴로지

크론잡과 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

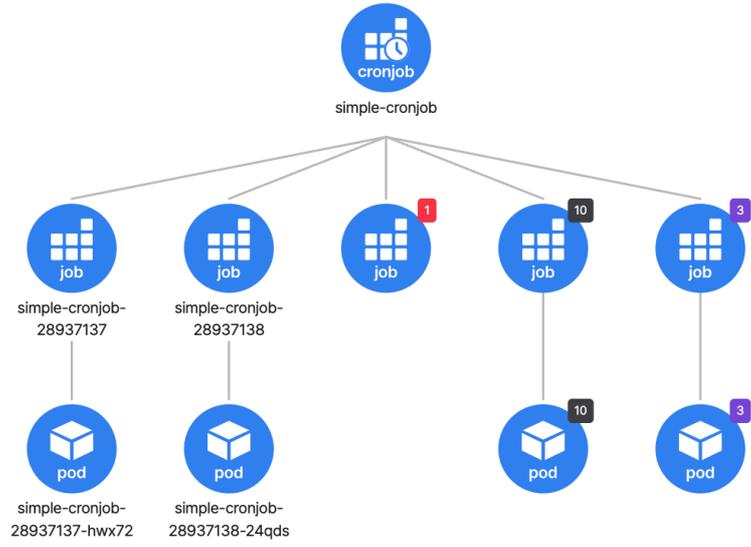
크론잡의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 크론잡, 잡, 파드

simple-cronjob

삭제

🔍 🔍 100% ↻



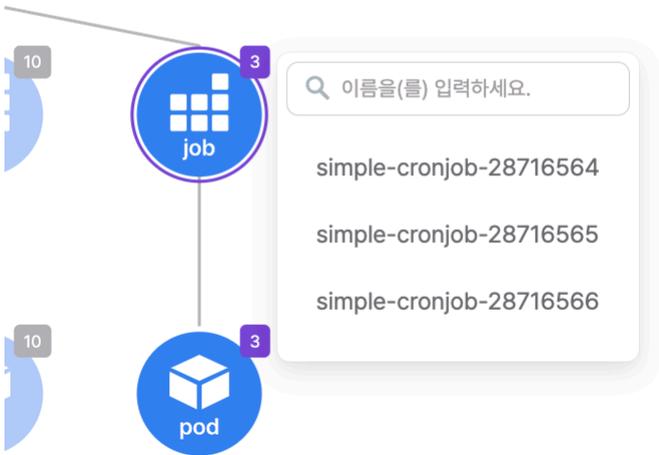
Running 상태인 잡과 파드는 기존 토폴로지와 동일하게 노드와 리소스 이름을 함께 출력한다.

Running 상태인 리소스를 제외하고는 상태에 따라 분류하여 총 리소스 개수를 보여주는 노드로 출력하며, 구분하는 기준과 색상은 아래와 같다.

12345 completed 12345 failed 12345 error

상태	설명
completed	잡이 성공적으로 완료된 상태
failed	잡이 실행되었지만 예기치 않은 이유로 작업이 완료되지 못한 상태
error	잡이 실행되기 전에 문제가 발생하여 실행되지 못한 상태

개수를 출력하는 노드를 클릭할 경우, 해당하는 상태의 잡 또는 파드 리스트를 드롭다운으로 출력하며 클릭할 경우 해당 리소스 화면으로 이동한다.



크론잡 생성

+ 크론잡 생성 을 선택하면 나타나는 화면에서 쿠버네티스 크론잡 리소스 정보를 입력하여 생성할 수 있다. 생성 시에는 FORM/YAML로 입력할 수 있으며 일반 설정과 컨테이너 설정을 하게 된다.

일반 설정

워크로드 > 크론잡

← 크론잡 목록 FORM YAML [크론잡 생성](#)

① 일반 설정 ② 컨테이너 설정

이름

네임스페이스

스케줄

재시작 정책

[다음: 컨테이너 설정 →](#)

항목	설명
이름	크론잡 이름
네임스페이스	크론잡이 생성될 네임스페이스
스케줄	크론 형식으로 작업이 실행될 시간을 정의
재시작 정책	크론잡이 생성하는 Pod가 실패할 경우 재시작 여부와 정책 설정 <ul style="list-style-type: none"> • OnFailure는 실패 시 재시작 • Never는 재시작하지 않음

컨테이너 설정

워크로드 > 크론잡

← 크론잡 목록

FORM YAML

크론잡 생성

- 1 일반 설정
- 2 컨테이너 설정

컨테이너

컨테이너 1 ?

name *

hello

image *

busybox

args

args 1

/bin/sh

삭제

args 2

-c

삭제

args 3

date; echo Hello Accordion

삭제

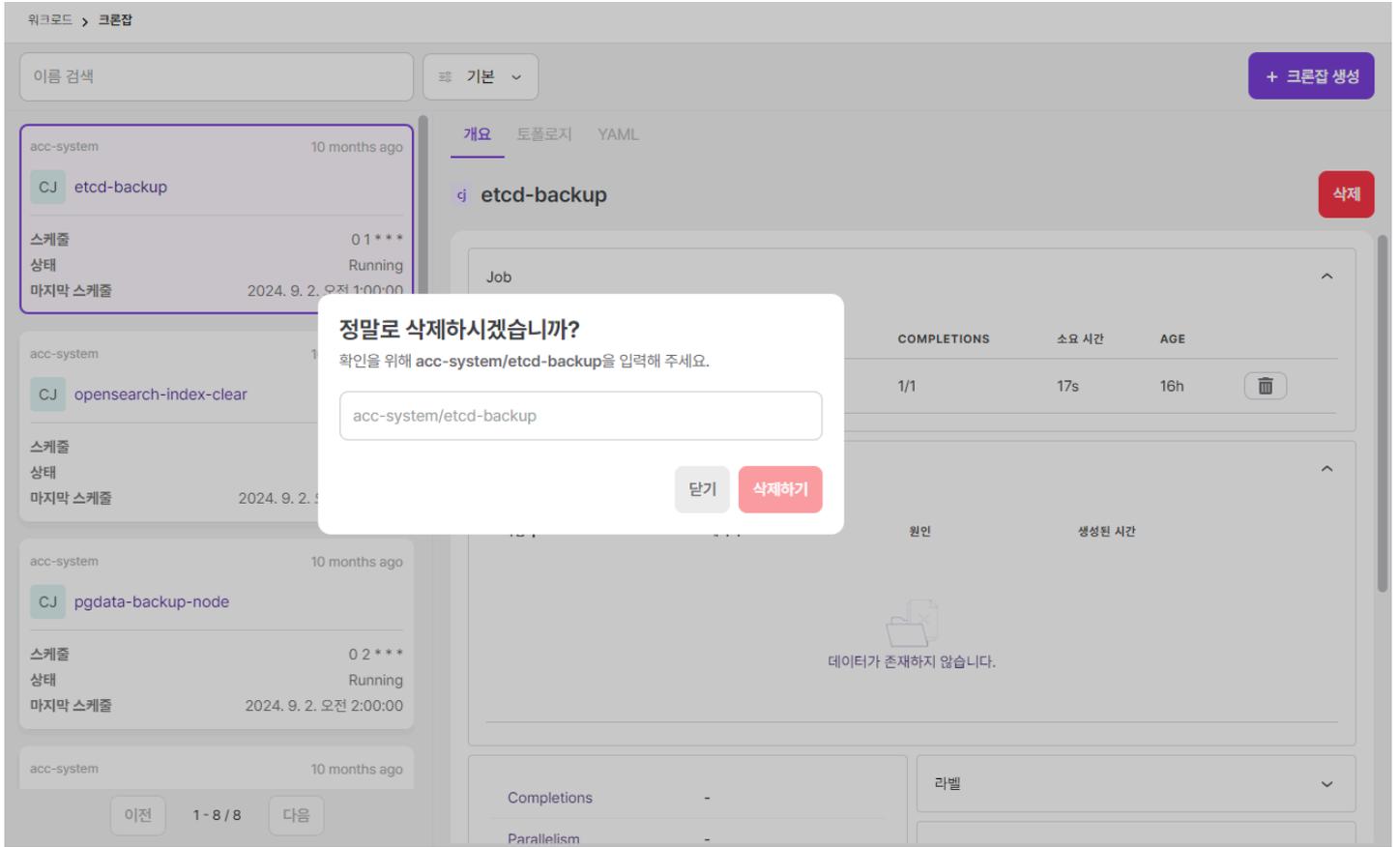
항목	설명
name	크론잡에서 실행될 컨테이너 이름
image	크론잡에서 실행될 컨테이너 이미지
args	컨테이너가 시작될 때 실행될 명령어

크론잡 수정

수정하려는 크론잡을 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

크론잡 삭제

삭제하려는 크론잡을 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 크론잡 이름을 입력하여 삭제한다.

4.2.7. 구성

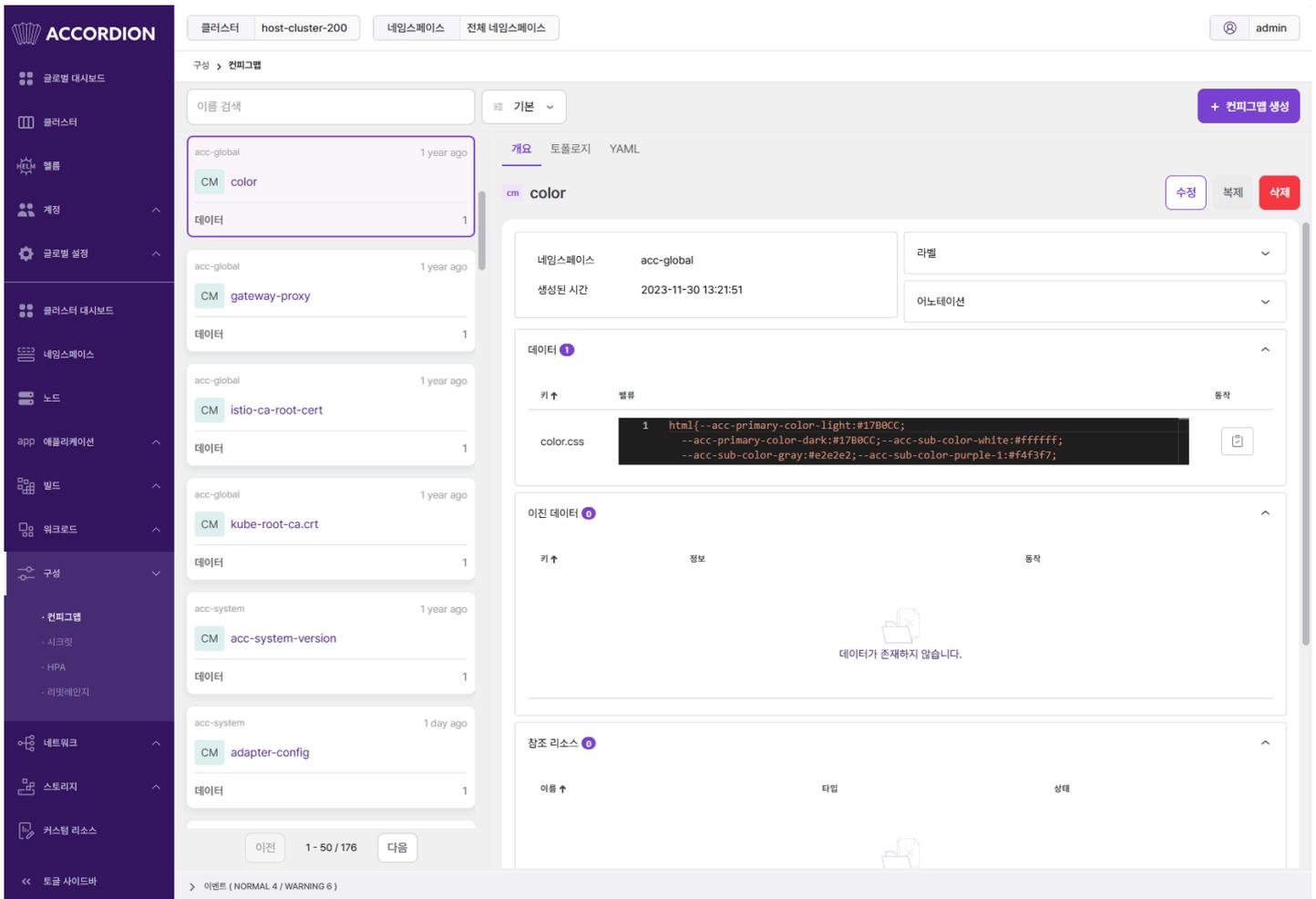
컨피그맵과 시크릿은 애플리케이션(또는 워크로드)가 정상적으로 실행되기 위해 필요한 구성요소 정보를 저장한다. 그 외에도 오토스케일 설정을 위한 HPA와 리소스 제한을 위한 리밋레인지 설정을 할 수 있다.

4.2.7.1. 컨피그맵

컨피그맵은 키-값 쌍으로 기밀이 아닌 데이터를 저장한다.

개요

배포된 컨피그맵의 세부 정보, 데이터, 이진 데이터, 참조 리소스, 이벤트를 제공한다.



세부 정보

컨피그맵에 대한 세부 정보를 제공한다.

네임스페이스	acc-global	라벨	▼
생성된 시간	2023-11-30 13:21:51	이노테이션	▼

데이터

컨피그맵의 데이터 정보를 제공하며 우측에 복사 버튼을 제공한다.

키 수	별명	동작
404	<pre> 4 <body>PAGE_NOT_FOUND !!!</body> 5 </html> 6 </pre>	 
4xx	<pre> 1 <!DOCTYPE html> 2 <html> 3 <head><title>PAGE_NOT_FOUND</title></head> </pre>	

이진 데이터

컨피그맵의 이진 데이터 정보를 제공하며 우측에 미리보기와 다운로드 버튼을 제공한다.

키 수	정보	동작
accordion3.png	image/png : 121041 Byte	 
filebeat.png	image/png : 23853 Byte	 

참조 리소스

컨피그맵의 참조 리소스 정보를 제공하며 클릭 시 해당 리소스로 이동된다.

이름 ↑	타입	상태
istio-egressgateway	Deployment	Ready (1/1)
istio-egressgateway-6584655bbf-xmg87	Pod	Running

이벤트 정보

컨피그맵에서 발생하는 이벤트 정보를 제공한다.

타입 ↑	메시지	원인	생성된 시간
Normal	Created	Created	4m

항목	설명
타입	이벤트 타입 <ul style="list-style-type: none"> Normal: 일반적인 작업으로 발생한 이벤트 Warning: 오류에 의해 발생한 이벤트
메시지	이벤트 메시지
원인	이벤트 발생 이유
생성된 시간	이벤트 생성 후 지난 시간

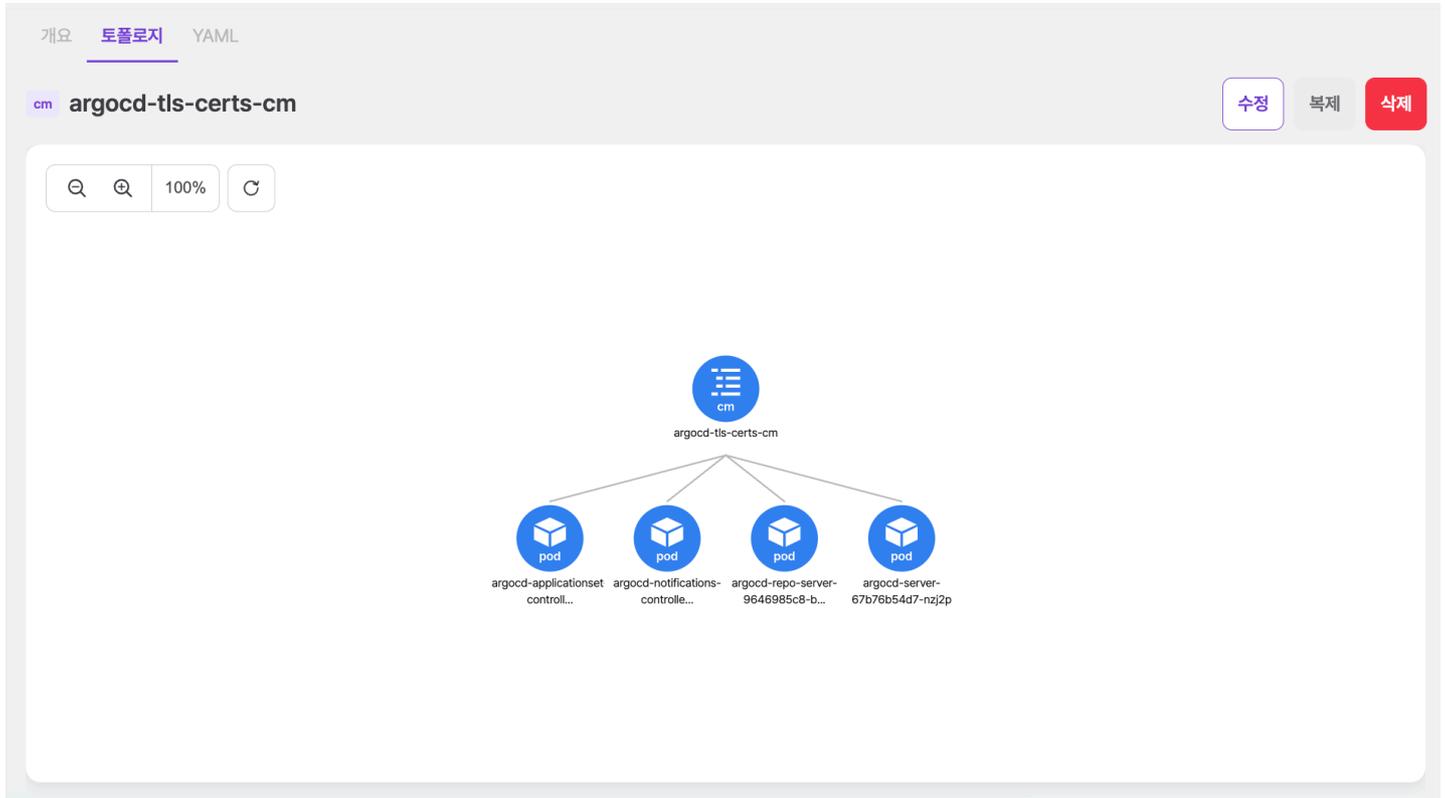
토폴로지

컨피그맵과 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

컨피그맵의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 컨피그맵
- 파드



컨피그맵 생성

+ 컨피그맵 생성 을 선택하면 나타나는 화면에서 쿠버네티스 컨피그맵 리소스 정보를 입력하여 생성할 수 있다. 생성 시에는 FORM/YAML로 입력할 수 있다.

구성 > 컨피그맵

← 컨피그맵 목록 FORM YAML [컨피그맵 생성](#)

이름

네임스페이스

Data

키	밸류	
	1	<input type="button" value="삭제"/>

컨피그맵 수정

수정하려는 컨피그맵을 선택하고 우측의 수정 버튼을 클릭하여 정보 수정 후 컨피그맵 수정 버튼을 클릭하여 반영한다.

구성 > 컨피그맵

← 컨피그맵 목록

FORM YAML

컨피그맵 수정

이름
scouter-agent

네임스페이스
aaa

Data

키	밸류	
scouter-agent.conf	<pre> 1 ##### Profile 2 ## Identifying header key of remote ip. (default: "") 3 ## eg) X-Forwarded-For </pre>	삭제

추가

Binary Data

키	밸류
데이터가 존재하지 않습니다.	

추가

컨피그맵 복제

컨피그맵을 다른 클러스터, 네임스페이스에 복제할 수 있다.

구성 > 컨피그맵

← 컨피그맵 목록

FORM YAML

컨피그맵 복제

클러스터
host-cluster-200

네임스페이스
네임스페이스

이름
color-copy

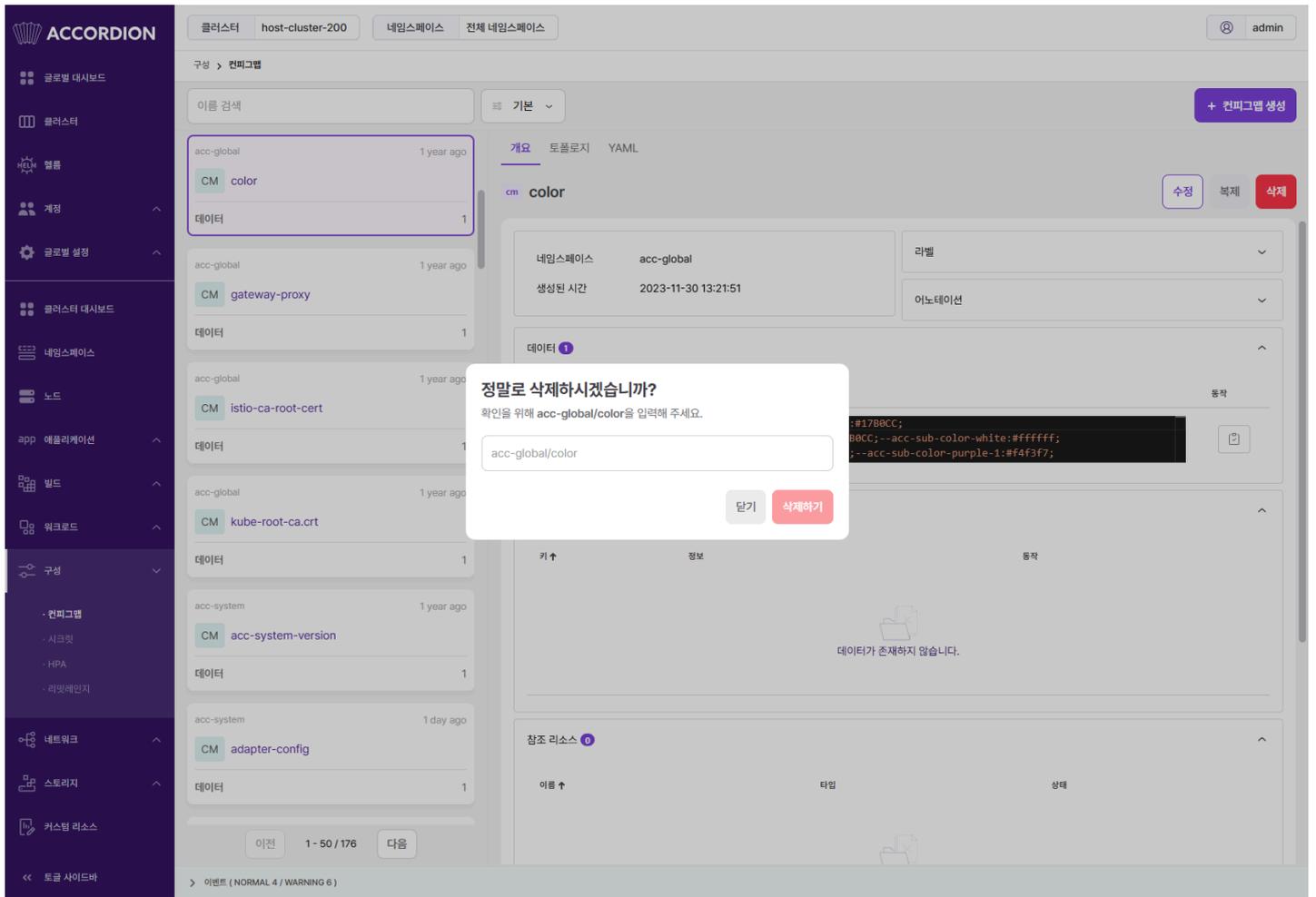
Data

키	밸류	
color.css	<pre> 1 html{--acc-primary-color-light:#17B0CC; --acc-primary-color-dark:#17B0CC;--acc-sub-color-white:#ffffff; --acc-sub-color-gray:#e2e2e2;--acc-sub-color-purple-1:#f4f3f7; </pre>	삭제

추가

컨피그맵 삭제

삭제하려는 컨피그맵을 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 컨피그맵 이름을 입력하여 삭제한다.

4.2.7.2. 시크릿

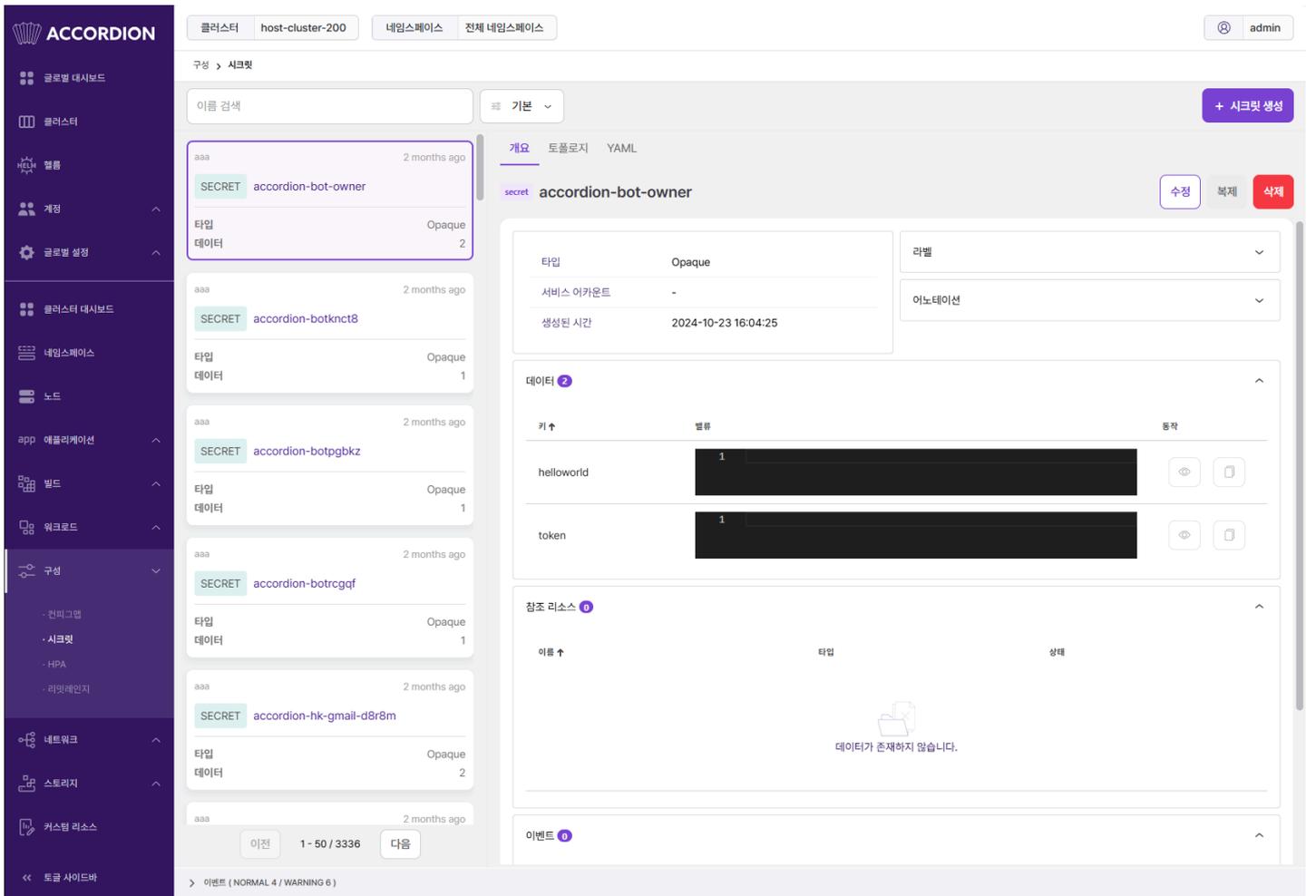
시크릿은 비밀번호, OAuth 토큰, Docker Registry , SSH 키와 같은 민감 정보를 저장하고 관리한다. 기밀 정보는 워크로드에 정의하는 것보다 안전하고 유연하다.

IMPORTANT

민감정보는 컨피그맵이 아닌 시크릿을 사용하여 안전하게 데이터를 저장해야한다. 시크릿에 저장된 정보는 조회 시 base64로 인코딩되어 표시된다.

개요

배포된 시크릿의 세부 정보, 데이터, 참조 리소스, 이벤트를 제공한다.



세부 정보

시크릿에 대한 세부 정보를 제공한다.

타입	kubernetes.io/service-account-token
서비스 어카운트	alert-apiserver
생성된 시간	2022-9-28 18:44:36

라벨	>
어노테이션	>

데이터

시크릿의 데이터 정보를 제공하며 우측에 base64디코딩 값 보기와 복사 버튼을 제공한다.

키 ↑	밸류	동작
ca.crt	LS0eLS1CRUdJTi0DRVJUSUZjO0FURS0eLS0eck1JSUWwRENDQWJpZ0F3SUJ6Z01KQURkdGQyMTc0Nm1RTUEwR0NTcUdTSWZzRFFkQkN3VUFNQ1V4RXpBUk3nT1YkQkFNTUNTdDFZbVZ5Ym1WMEpYTXdIaGh0TWpJd09USTRMRG55TVRNMVdoY050REl3T1R0ek1Ea311NVE0xV2	<input type="button" value="👁️"/> <input type="button" value="📄"/>
namespace	YWNjLWdsb2JhbA==	<input type="button" value="👁️"/> <input type="button" value="📄"/>

참조 리소스

컨피그맵과 내용이 동일하기 때문에 컨피그맵 참조 리소스를 참고한다.

이벤트 정보

컨피그맵과 내용이 동일하기 때문에 컨피그맵 이벤트 정보를 참고한다.

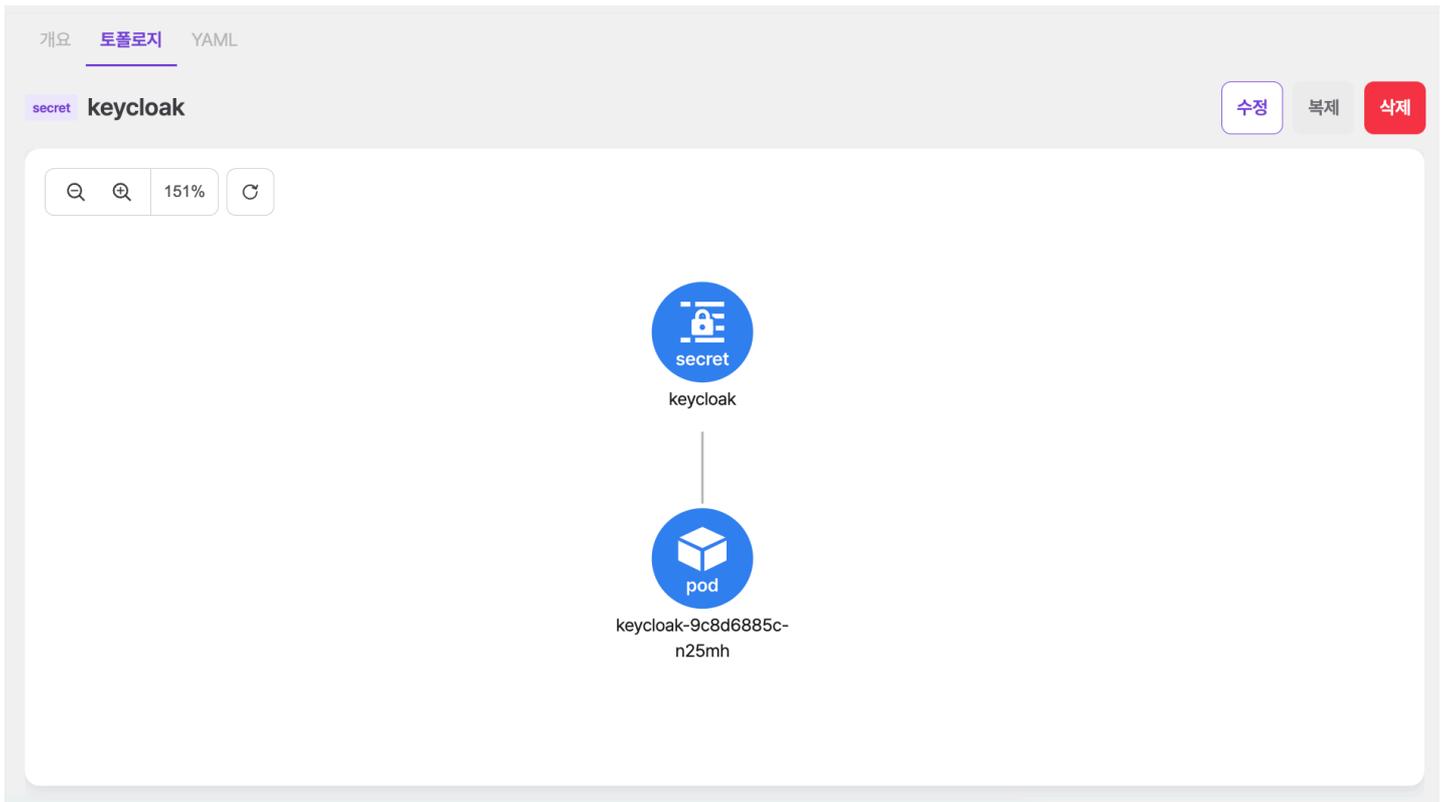
토폴로지

시크릿과 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

시크릿의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 시크릿
- 파드



시크릿 생성

+ 시크릿 생성 을 선택하면 나타나는 화면에서 쿠버네티스 시크릿 리소스 정보를 입력하여 생성할 수 있다. 생성 시에는 FORM/YAML로 입력할 수 있다.

구성 > 시크릿

← 시크릿 목록 FORM YAML 시크릿 생성

이름

네임스페이스

시크릿 타입

Data

키	밸류	
	1	<input type="button" value="삭제"/>

시크릿 수정

수정하려는 시크릿을 선택하고 우측의 수정 버튼을 클릭하여 정보 수정 후 시크릿 수정 버튼을 클릭하여 반영한다.

구성 > 시크릿

← 시크릿 목록 FORM YAML 시크릿 수정

이름

네임스페이스

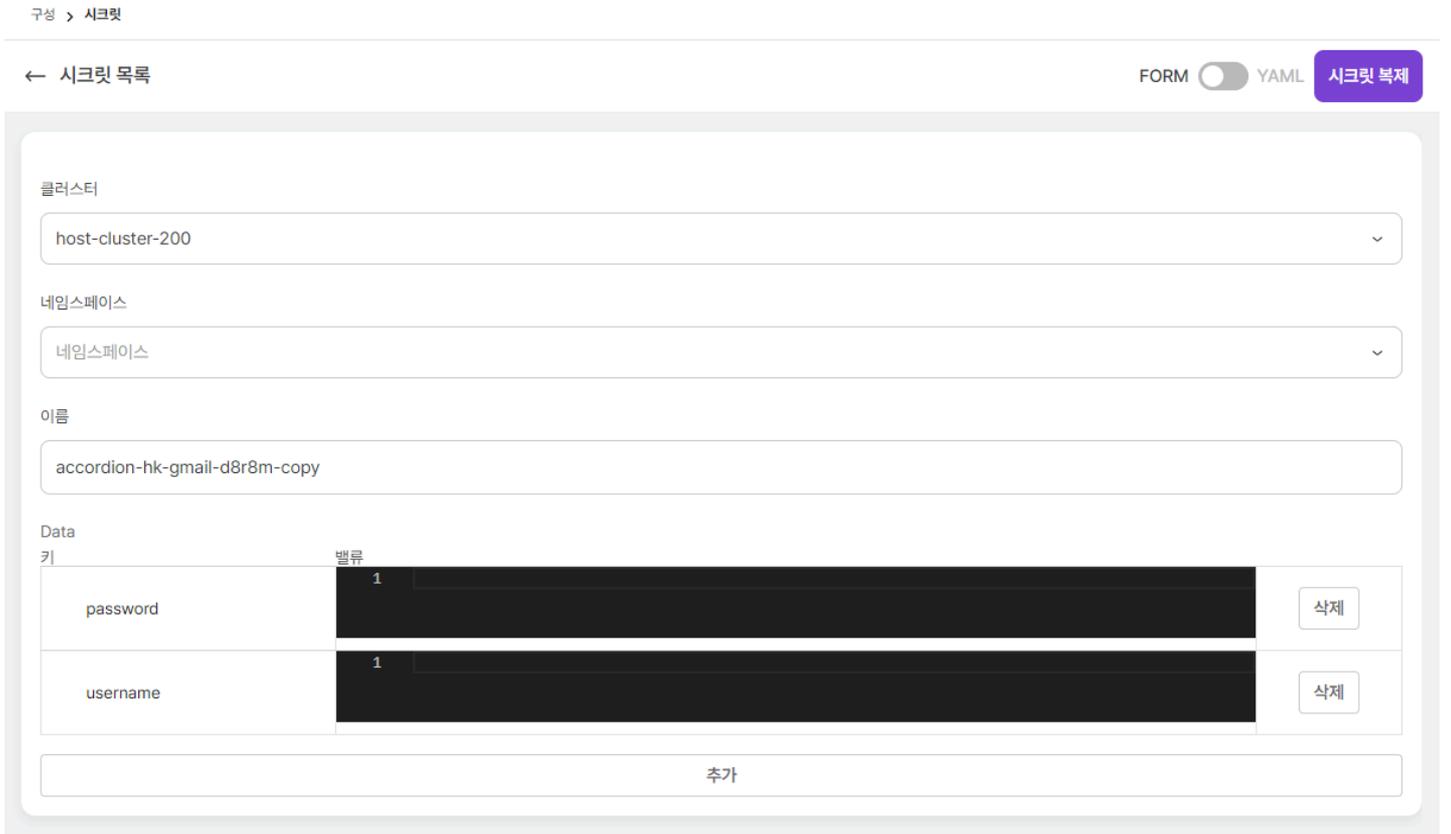
시크릿 타입

Data

키	밸류	
helloworld	1	<input type="button" value="삭제"/>
token	1	<input type="button" value="삭제"/>

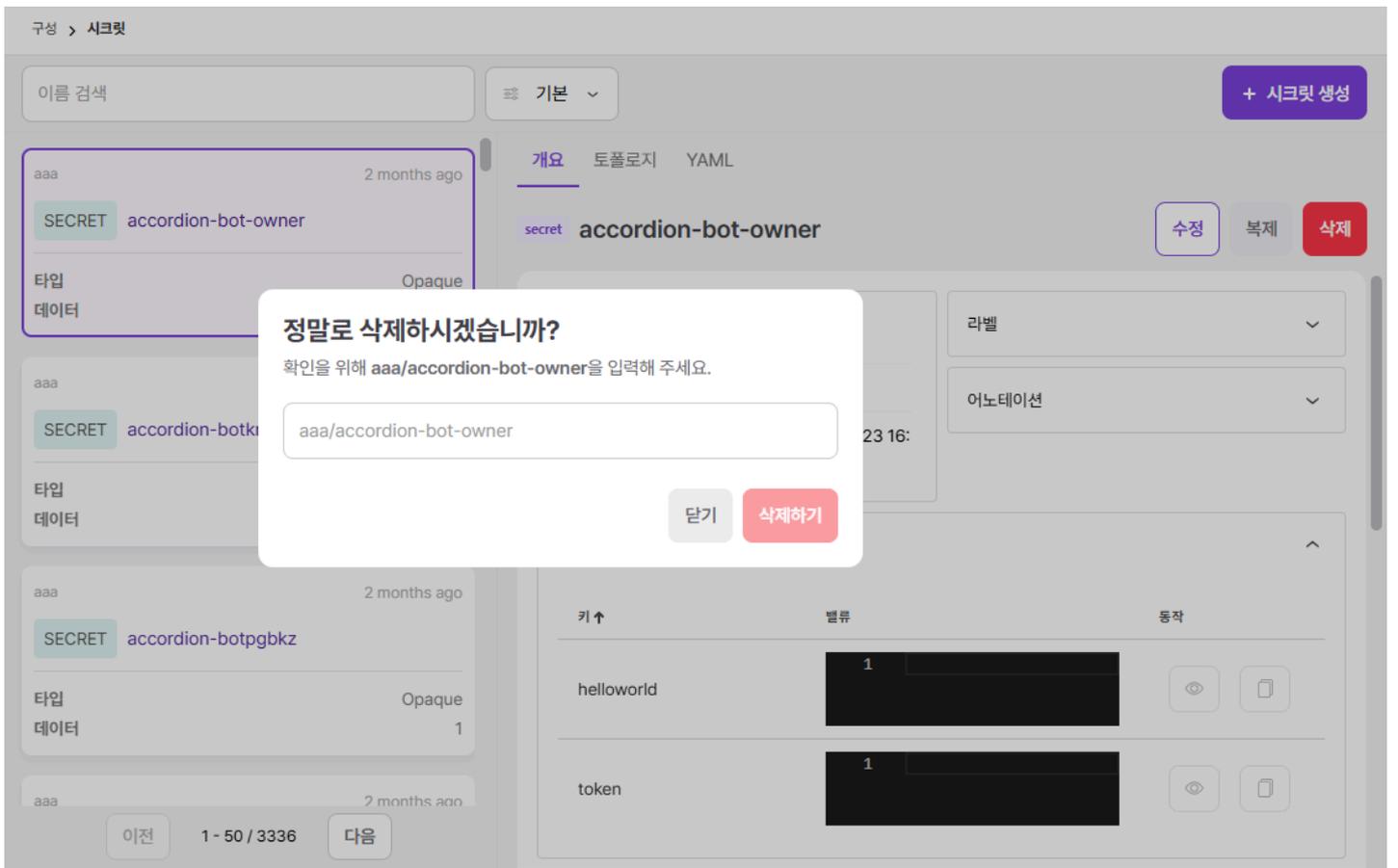
시크릿 복제

시크릿을 다른 클러스터, 네임스페이스에 복제할 수 있다.



시크릿 삭제

삭제하려는 시크릿을 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 시크릿 이름을 입력하여 삭제한다.

4.2.7.3. HPA

HPA는 파드를 수평적으로 확장/관리하는 쿠버네티스 리소스이다. 리소스 메트릭을 기준으로 파드를 확장한다.

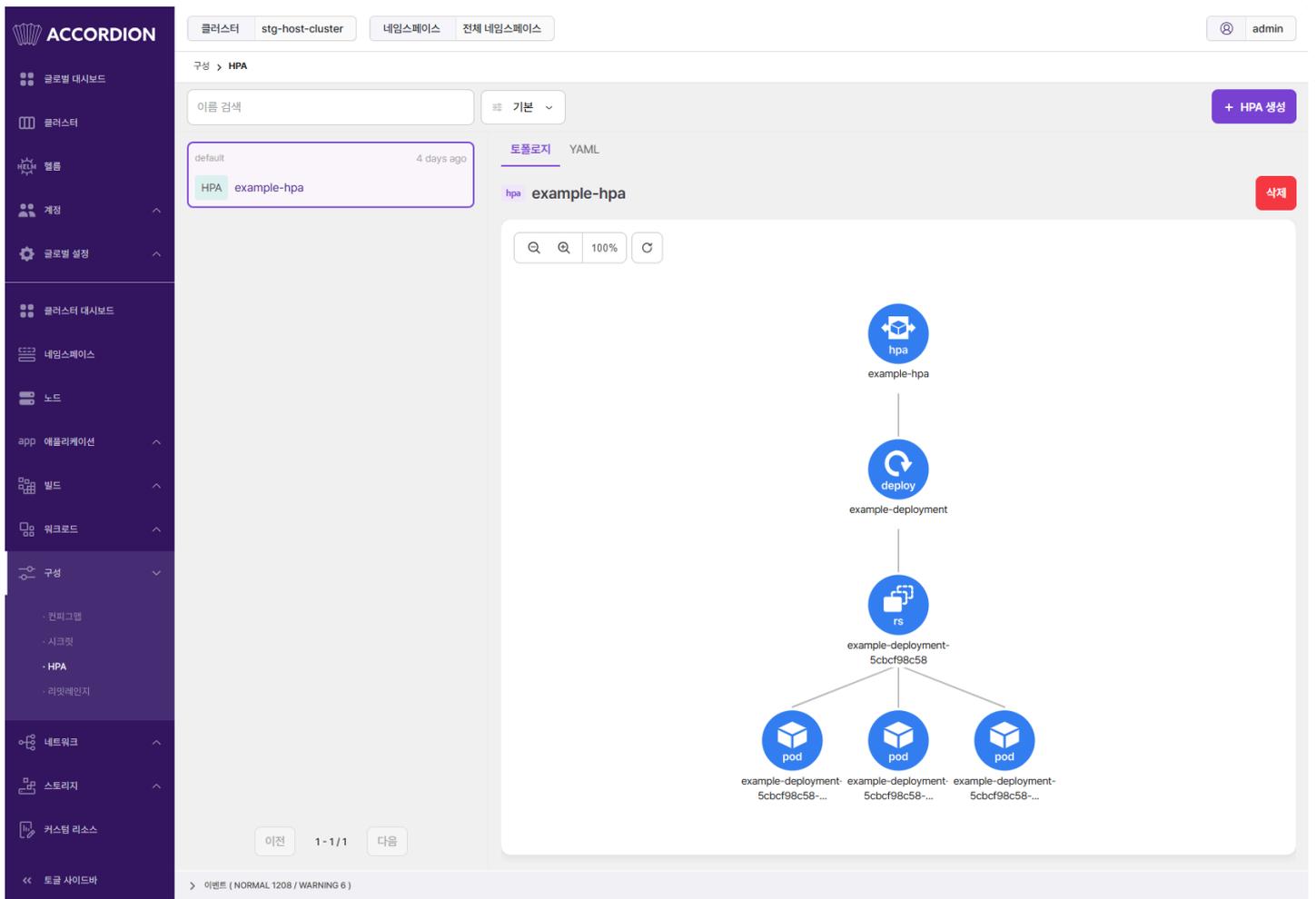
토폴로지

HPA과 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

HPA의 토폴로지에서 출력되는 리소스는 다음과 같다.

- HPA
- 디플로이먼트, 레플리카셋, 스테이트풀셋
- 파드



HPA 생성

+ HPA 생성 을 선택하면 나타나는 화면에서 쿠버네티스 HPA 리소스 정보를 입력하여 생성할 수 있다.

구성 > HPA

← HPA 목록

HPA 생성

```

1  apiVersion: autoscaling/v2
2  kind: HorizontalPodAutoscaler
3  metadata:
4    name: example-hpa
5    namespace: default
6  spec:
7    scaleTargetRef:
8      apiVersion: apps/v1
9      kind: Deployment
10     name: example-deployment
11   minReplicas: 1
12   maxReplicas: 10
13   metrics:
14     - type: Resource
15       resource:
16         name: cpu
17         target:
18           type: Utilization
19           averageUtilization: 50
20

```

HPA 수정

수정하려는 HPA를 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

구성 > HPA

이름 검색

기본

+ HPA 생성

default 4 days ago

HPA example-hpa

토폴로지 YAML

hpa example-hpa

수정

삭제

```

1  kind: HorizontalPodAutoscaler
2  apiVersion: autoscaling/v1
3  metadata:
4    name: example-hpa
5    namespace: default
6    uid: ce60311c-6464-409f-b60c-a3eb1e7a682b
7    resourceVersion: "50636107"
8    creationTimestamp: "2025-01-03T01:10:57Z"
9    annotations:
10     autoscaling.alpha.kubernetes.io/conditions: >-
11       [{"type":"AbleToScale","status":"True","lastTransitionTime":"2025-01-03T01:10:57Z","reason":"HPA controller was able to update the target scale to 2"}, {"type":"ScalingActive","status":"True","lastTransitionTime":"2025-01-03T01:10:57Z","reason":"HPA was able to successfully calculate a replica count from cpu resource utilization (percentage of request)"}, {"type":"ScalingLimited","status":"False","lastTransitionTime":"2025-01-03T01:10:57Z","reason":"desired count is within the acceptable range"}]
12     autoscaling.alpha.kubernetes.io/current-metrics: >-
13       [{"type":"Resource","resource":{"name":"cpu","currentAverageUtilization":50}}]
14

```

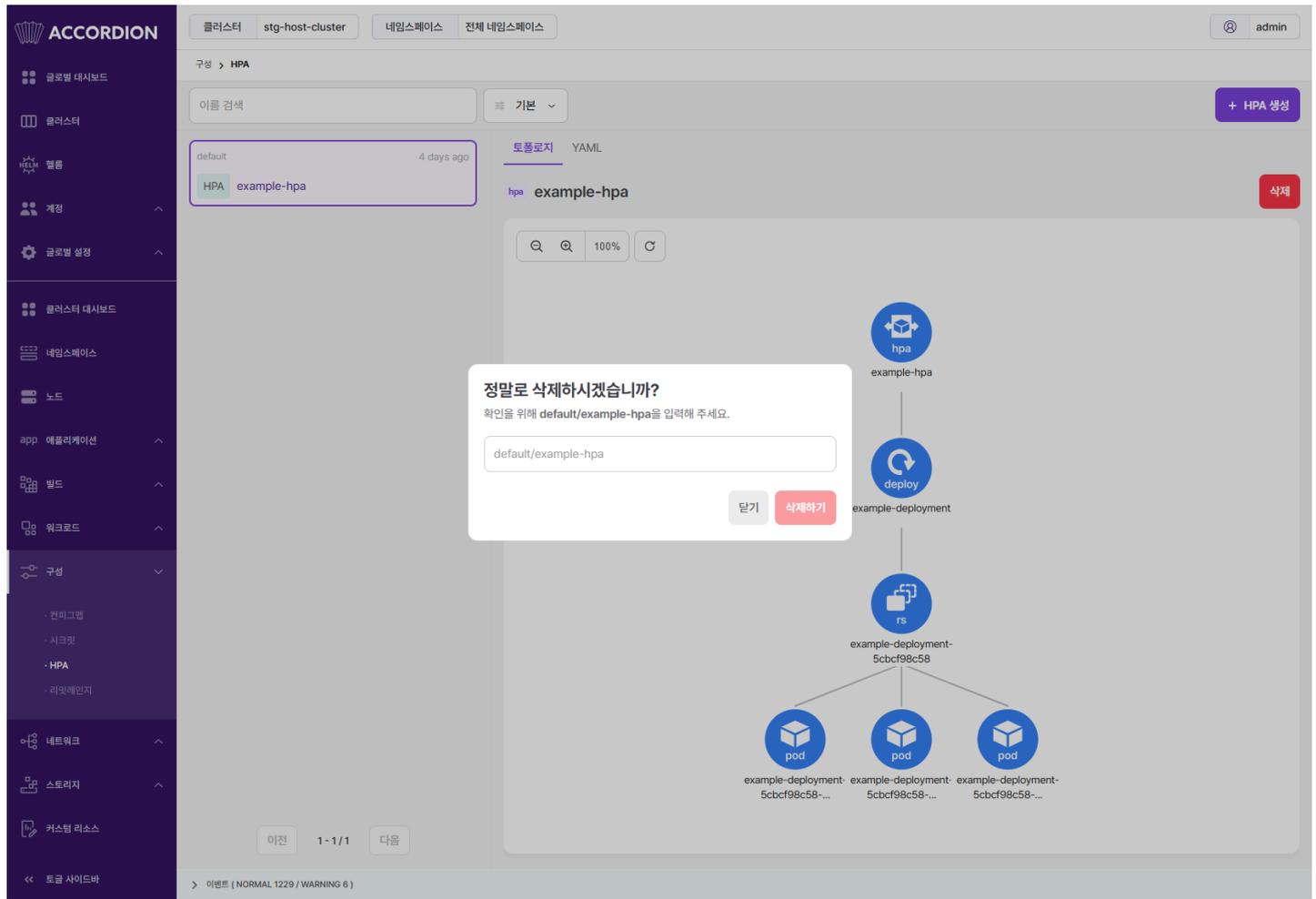
이전

1 - 1 / 1

다음

HPA 삭제

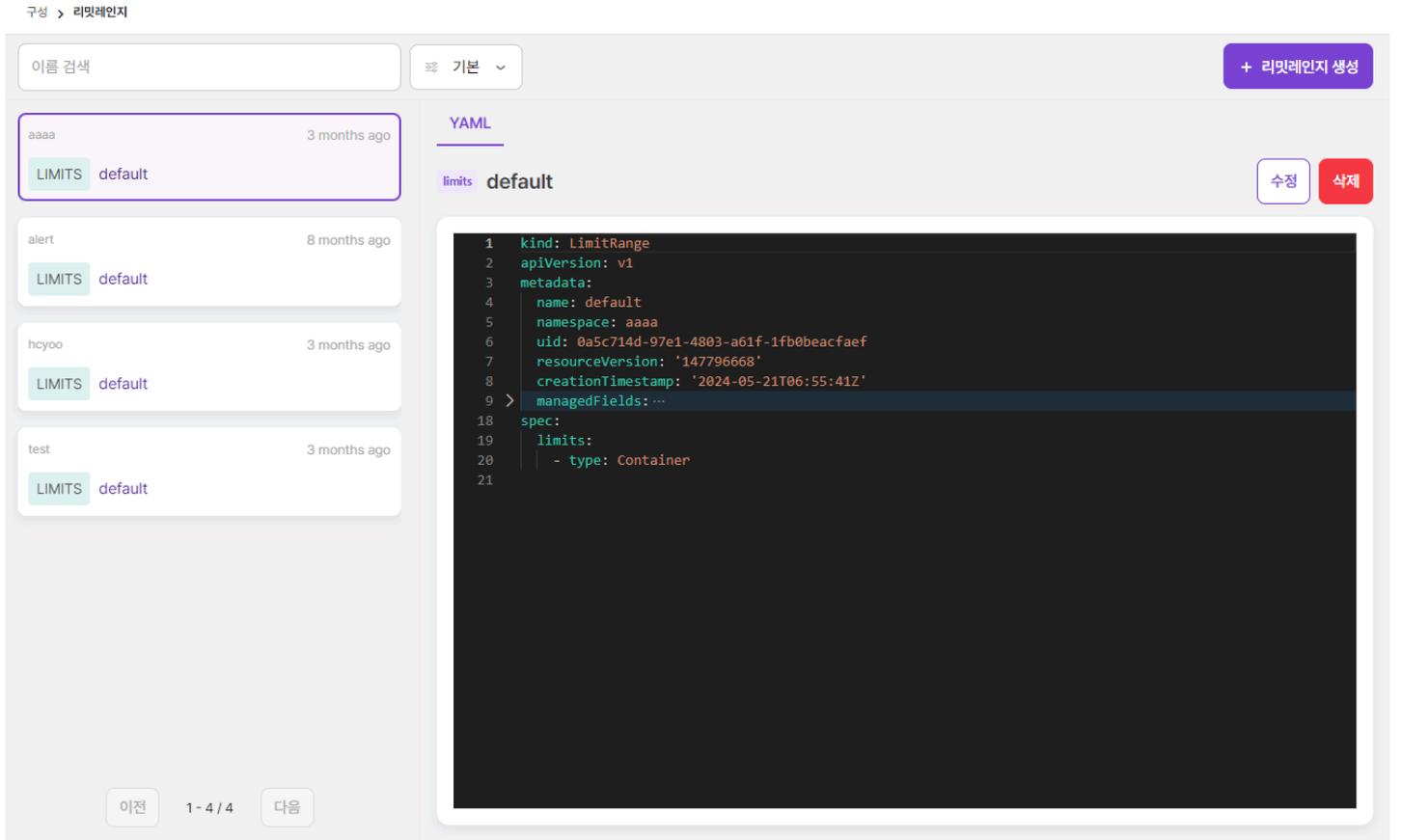
삭제하려는 HPA를 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 HPA 이름을 입력하여 삭제한다.

4.2.7.4. 리밋레인지

리밋레인지는 네임스페이스에서 파드(또는 컨테이너)에 대한 리소스 할당을 제한한다. 파드(또는 컨테이너)별 최소 및 최대 시스템 리소스(CPU 또는 메모리) 사용량을 지정하거나 스토리지클래스별 최소/최대 스토리지 요청 등을 지정한다.



리밋레인지 생성

+ 리밋레인지 생성 을 선택하면 나타나는 화면에서 쿠버네티스 리밋레인지 리소스 정보를 입력하여 생성할 수 있다.

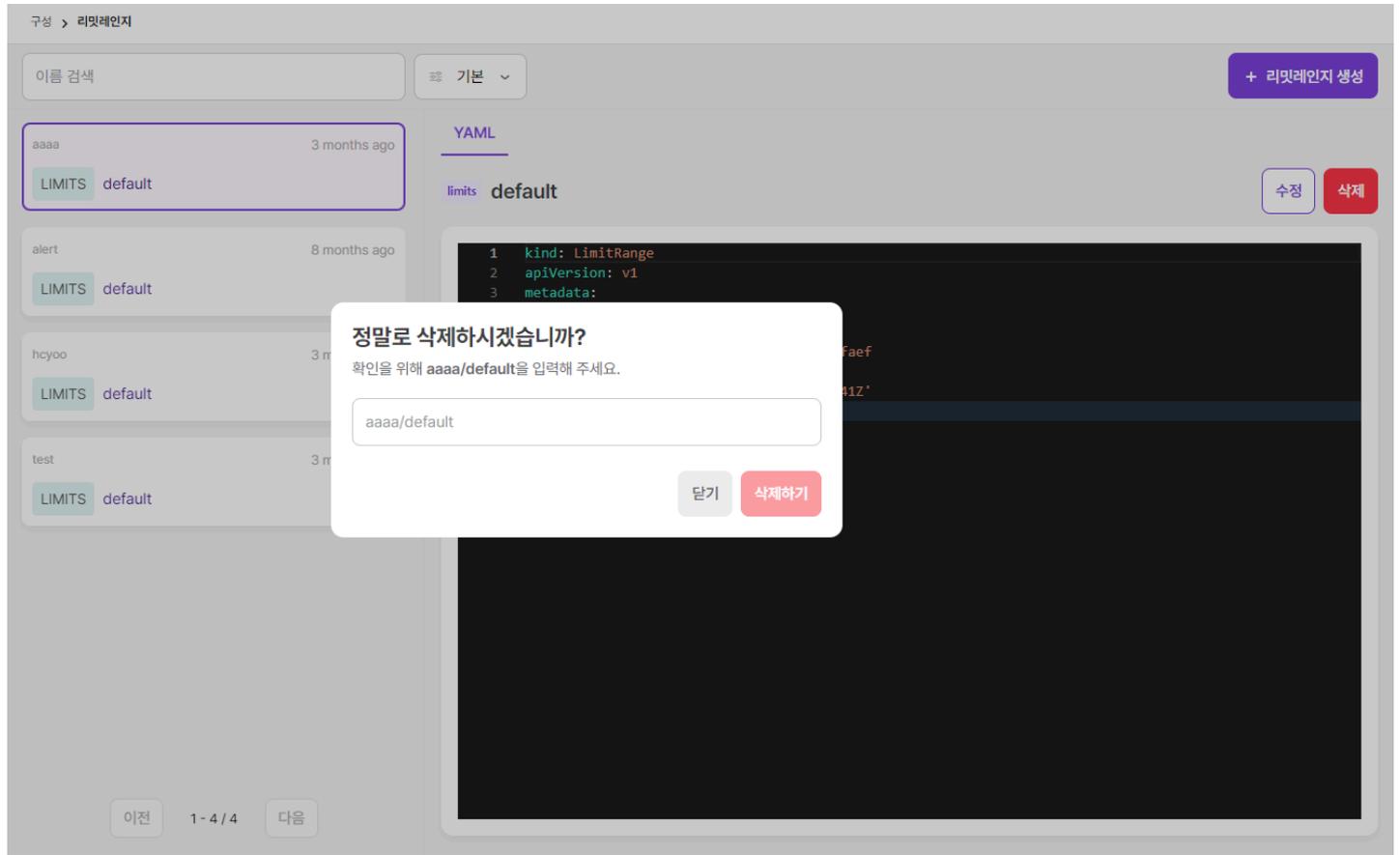
```
1  apiVersion: v1
2  kind: LimitRange
3  metadata:
4    name: example-limit-range
5    namespace: default
6  spec:
7    limits:
8      - default:
9          memory: "512Mi"
10         defaultRequest:
11             memory: "256Mi"
12         max:
13             memory: "1Gi"
14         min:
15             memory: "256Mi"
16         type: Container
17
```

리밋레인지 수정

수정하려는 리밋레인지를 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

리밋레인지 삭제

삭제하려는 리밋레인지를 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 리밋레인지 이름을 입력하여 삭제한다.

4.2.8. 네트워크

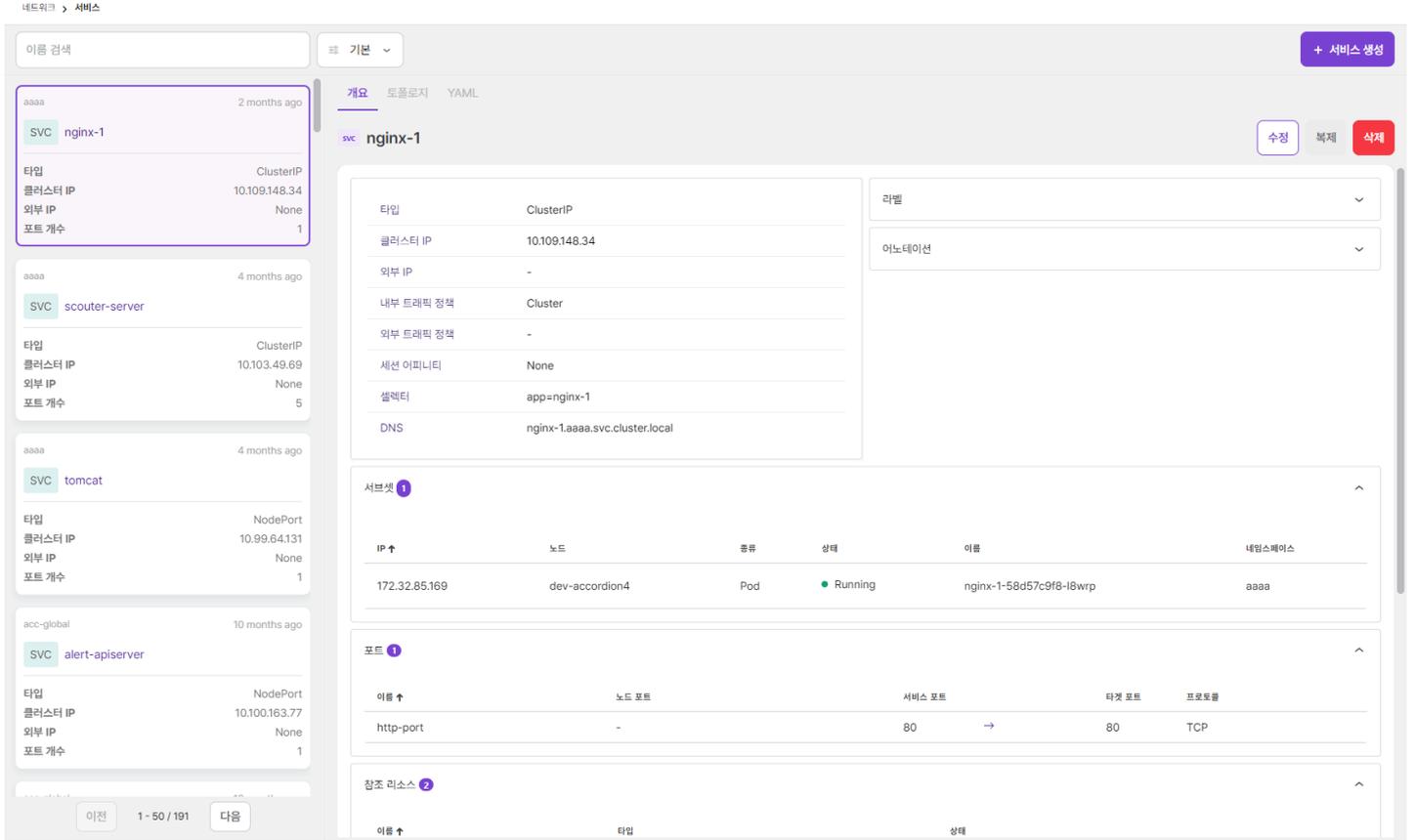
워크로드가 통신하기 위한 네트워크 설정을 관리한다.

4.2.8.1. 서비스

애플리케이션 및 워크로드의 파드 집합에 단일 DNS를 부여하고 로드밸런싱을 수행한다.

개요

서비스의 세부 정보, 서브셋, 포트, 참조 리소스, 이벤트를 제공한다.



세부 정보

서비스에 대한 세부 정보를 제공한다.

타입	NodePort
클러스터 IP	10.107.11.2
외부 IP	-
내부 트래픽 정책	Cluster
외부 트래픽 정책	Cluster
세션 어피니티	None
셀렉터	app=alert-apiserver
DNS	alert-apiserver.acc-global.svc.cluster.local

라벨

어노테이션

```
kubectcl.kubernetes.io/last-applied-configuration={\"apiVersion\":\"v1\",\"kind\":\"Service\",\"metadata\":{\"annotations\":{},\"name\":\"alert-...
```

서브셋

서비스에 매핑된 엔드포인트 정보에 따라 파드 정보를 제공하며 클릭 시 해당 리소스로 이동된다.

IP ↑	노드	종류	상태	이름	네임스페이스
172.32.24.43	acc-node2	Pod	Running	alert-apiserver-b876cfb65-wm5r6	acc-global

포트

서비스의 포트 정보 및 노드 포트에 접속할 수 있는 버튼을 제공한다.

포트 1				
이름 ↑	노드 포트	서비스 포트	타겟 포트	프로토콜
	31788	80	8080	TCP

참조 리소스

서비스의 참조 리소스 정보를 제공하며 클릭 시 해당 리소스로 이동된다.

참조 리소스 1		
이름 ↑	타입	상태
ac2-1351	Catalog	● Succeeded / 21

이벤트 정보

서비스에서 발생하는 이벤트 정보를 제공한다.

이벤트 6			
타입 ↑	메시지	원인	생성된 시간
Normal	Scheduled for sync	Sync	9d
Normal	Scheduled for sync	Sync	9d
Normal	Scheduled for sync	Sync	9d

항목	설명
타입	이벤트 타입 <ul style="list-style-type: none"> • Normal: 일반적인 작업으로 발생된 이벤트 • Warning: 오류에 의해 발생된 이벤트
메시지	이벤트 메시지
원인	이벤트 발생 이유
생성된 시간	이벤트 생성 후 지난 시간

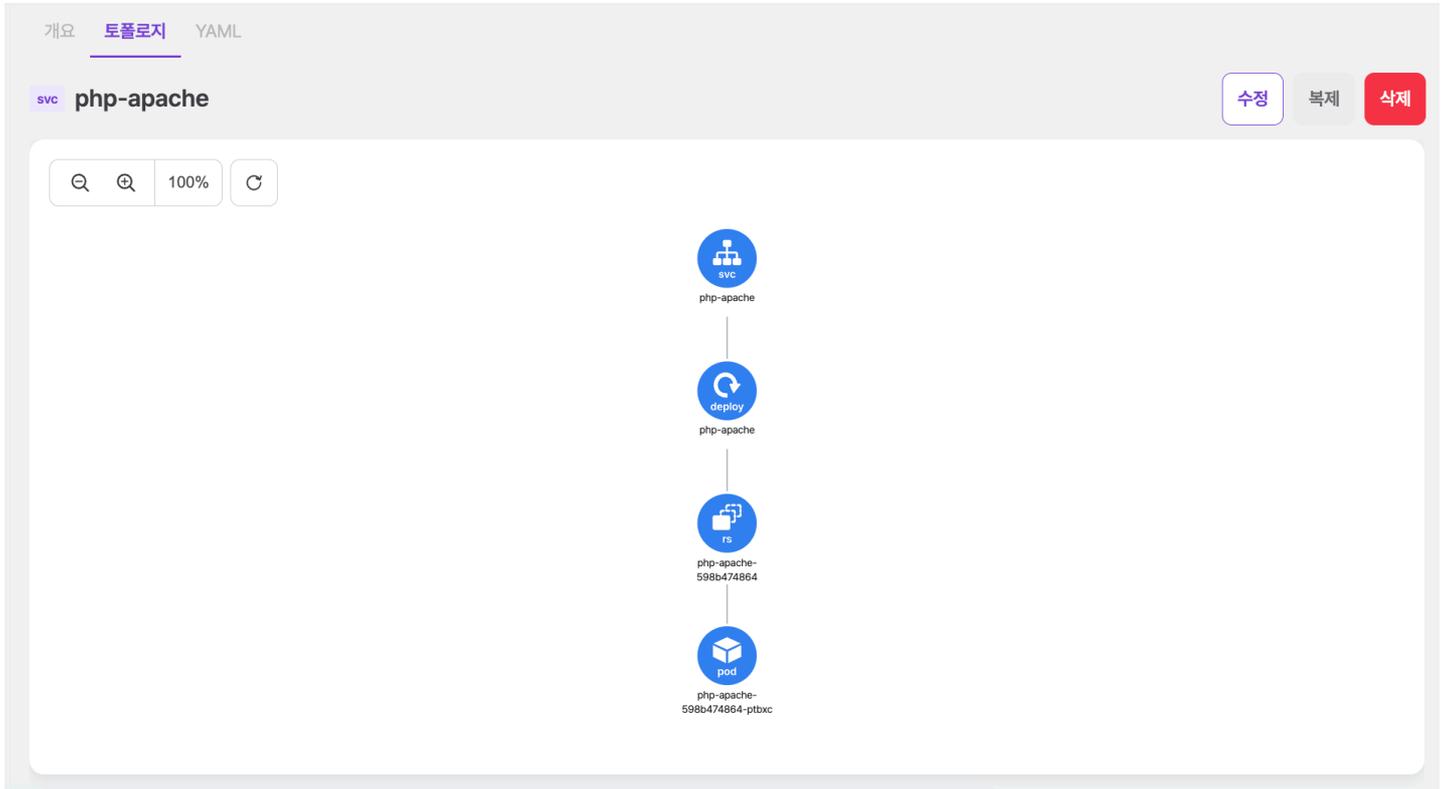
토폴로지

서비스와 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

서비스의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 디플로이먼트, 레플리카셋, 데몬셋, 스테이트풀셋, 파드
- 서비스



서비스 생성

+ 서비스 생성 을 선택하면 나타나는 화면에서 쿠버네티스 서비스 리소스 정보를 입력하여 생성할 수 있다. 생성 시에는 FORM/YAML로 입력할 수 있다.

← 서비스 목록 FORM YAML 서비스 생성

이름

네임스페이스

셀렉터

키	밸류	
		추가

서비스 타입

클러스터 IP

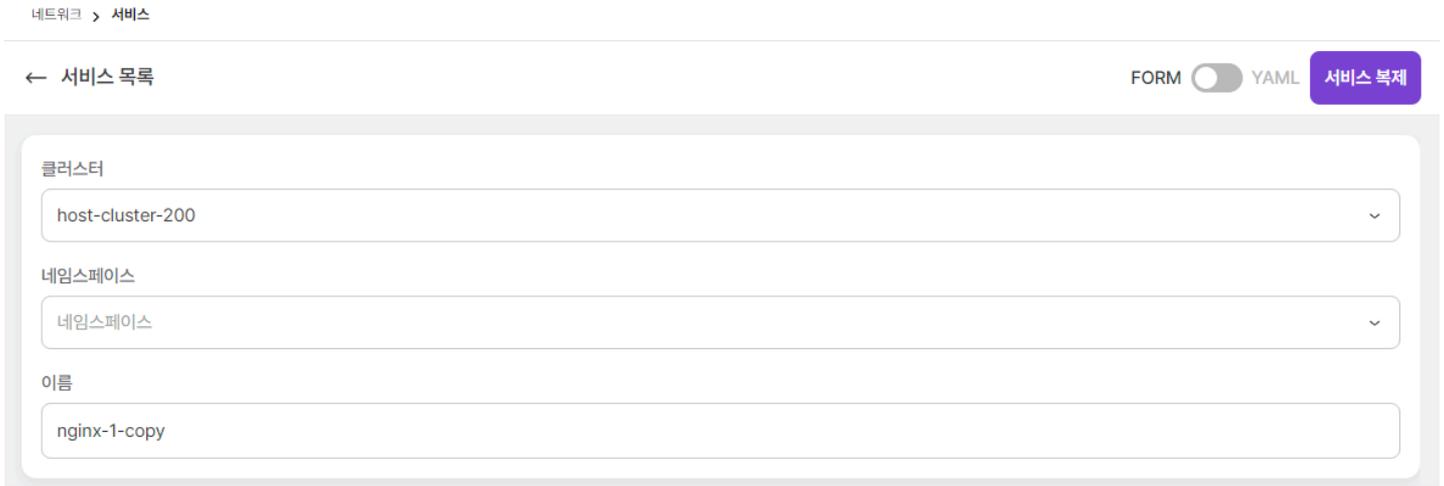
포트명	포트	프로토콜	타겟 포트	
				추가

서비스 수정

수정하려는 서비스를 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

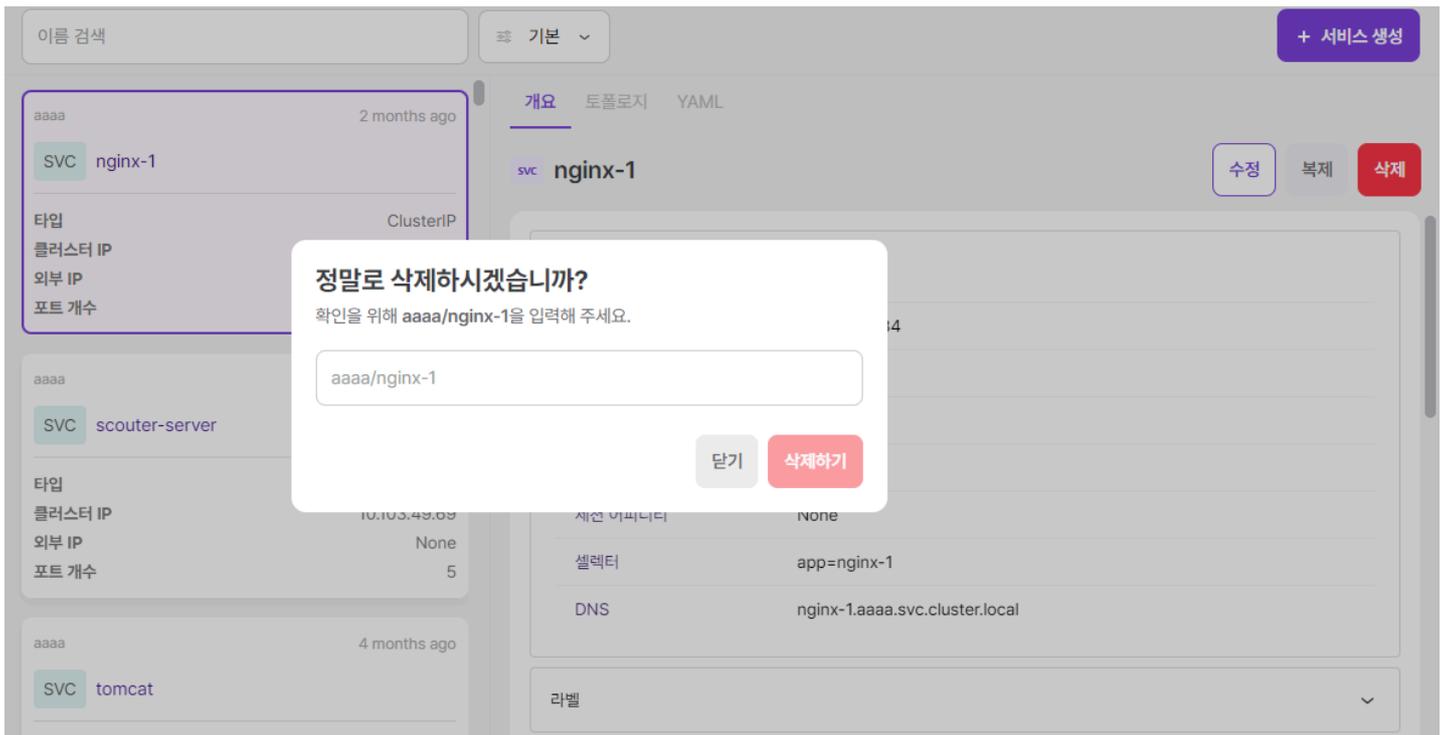
서비스 복제

서비스를 다른 클러스터, 네임스페이스에 복제할 수 있다.



서비스 삭제

삭제하려는 서비스를 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 서비스 이름을 입력하여 삭제한다.

4.2.8.2. 인그레스

인그레스는 클러스터 외부에서 클러스터 내부 서비스로 HTTP와 HTTPS 경로를 노출한다. 인그레스는 외부에서 서비스로 접속이 가능한 URL, 로드 밸런싱 트래픽, SSL/TLS 종료 그리고 이름-기반의 가상 호스팅을 제공하도록 구성할 수 있다.

개요

인그레스의 세부 정보, 라우트 규칙, 참조 리소스, 이벤트를 제공한다.

네트워크 > 인그레스

이름 검색

기본

+ 인그레스 생성

aaaa 2 months ago
ING jystore
호스트 jystore.com

acc-global 9 months ago
ING gateway-ing
호스트 accordion.com

dhcho-apm 6 days ago
ING dhcho-apm-ingress
호스트 dhcho-jpetstore.com

dhcho 4 days ago
ING dhcho-apm-ingress-copy
호스트 dhcho-jpetstore.com

hjkim 5 months ago
ING deploy-test
호스트 devkube.example.com

hwpark-1 3 months ago
ING hwpark-1-ingress
호스트 hwpark-1-ingress.test.local1

이전 1 - 29 / 29 다음

개요 토폴로지 YAML

ing deploy-test 수정 삭제

주소 10.20.200.204, 10.20.200.205
Default Backend <default>
인그레스 클래스 명 nginx
생성된 시간 2024-3-12 16:58:43

라벨

어노테이션

라우트 규칙

호스트	TLS	경로 유형	경로	백엔드	백엔드 IP	포트
devkube.example.com		Prefix	/ping	deploy-test	172.32.64.245	8080

참조 리소스

이름 ↑	타입	상태
deploy-test	Catalog	Failed / 7
deploy-test	Deployment	Ready (1/1)

이벤트

타입 ↑	메시지	확인	생성된 시간
------	-----	----	--------

세부 정보

인그레스에 대한 세부 정보를 제공한다.

주소 10.20.200.241
Default Backend nginx-errors:80 (172.32.139.173)
인그레스 클래스 명 user-ingress-class
생성된 시간 2022-11-07 15:58:32

라벨

데이터가 존재하지 않습니다.

어노테이션

키 ↑	값
nginx.ingress.kubernetes.io/backend-protocol	HTTP
nginx.ingress.kubernetes.io/custom-http-errors	404,415,504
nginx.ingress.kubernetes.io/default-backend	nginx-errors
nginx.ingress.kubernetes.io/rewrite-target	/

라우트 규칙

호스트	TLS	경로 유형	경로	백엔드	백엔드 IP	포트
ac2-1467-accordion.s	fileupload-secret2	Prefix	/	ac2-1467	172.32.24.46	8080

항목	설명
호스트	네트워크 요청이 들어오는 호스트 명
TLS	https 설정을 위한 시크릿 정보
경로 유형	<ul style="list-style-type: none"> ImplementationSpecific : IngressClass의 설정에 따름 Exact : url경로를 정확하게 일치 Prefix : /로 분할된 url 경로 접두사를 기반으로 일치
경로	요청의 경로
백엔드	호스트와 경로에 연결된 서비스
백엔드 IP	서비스의 IP
포트	서비스의 포트

참조 리소스

서비스와 내용이 동일하기 때문에 서비스 참조 리소스를 참고한다.

이벤트 정보

서비스와 내용이 동일하기 때문에 서비스 이벤트 정보를 참고한다.

토폴로지

인그레스와 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

인그레스의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 인그레스, 서비스
- 파드
- 시크릿

개요 **토폴로지** YAML

ing gateway-ing 수정 복제 삭제

🔍 🔍 99% ↻

인그레스 생성

+ 인그레스 생성 을 선택하면 나타나는 화면에서 쿠버네티스 인그레스 리소스 정보를 입력하여 생성할 수 있다. 생성 시에는 FORM/YAML로 입력할 수 있다.

← 인그레스 목록 FORM YAML 인그레스 생성

이름*

네임스페이스*

라벨

어노테이션

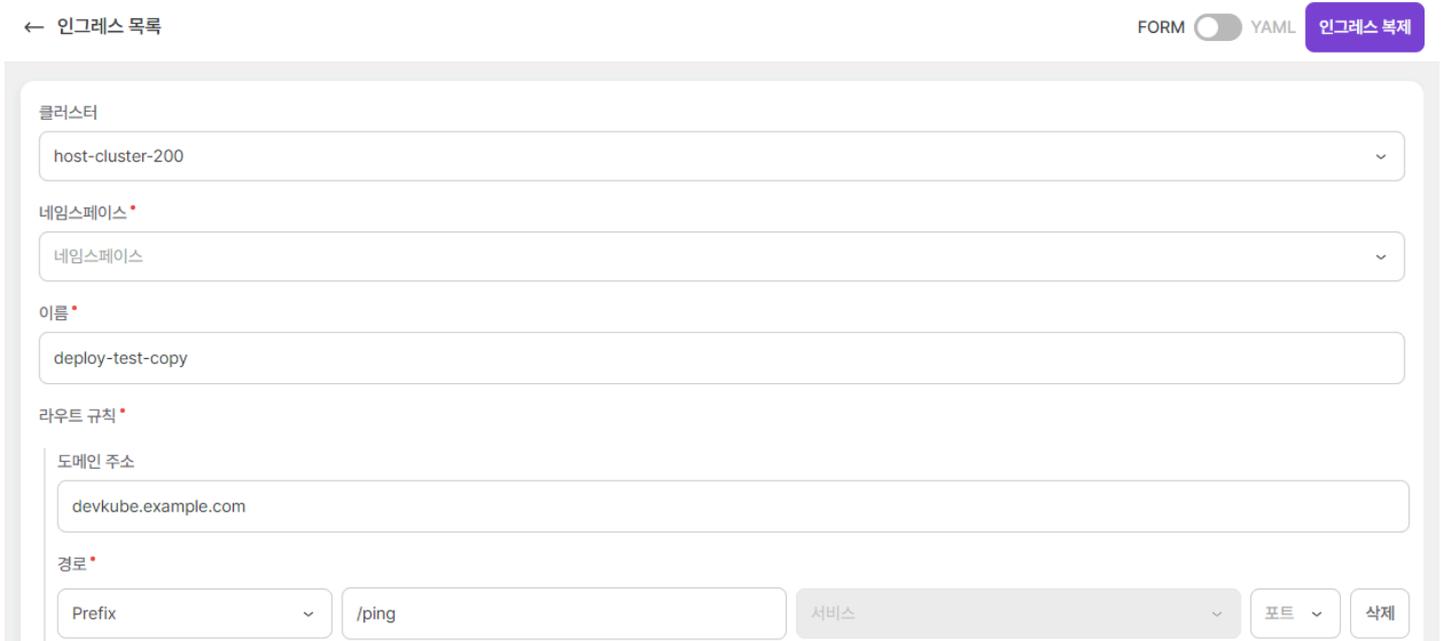
인그레스 클래스 명

인그레스 수정

수정하려는 인그레스를 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

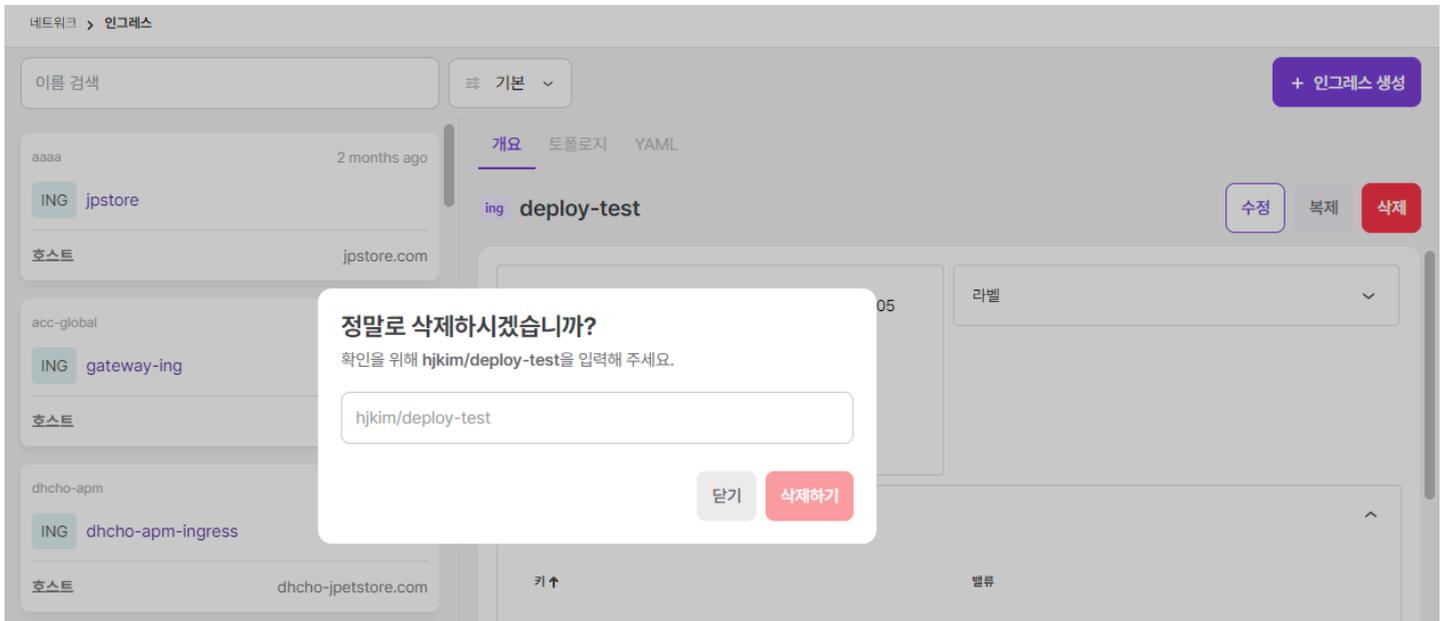
인그레스 복제

인그레스를 다른 클러스터, 네임스페이스에 복제할 수 있다.



인그레스 삭제

삭제하려는 인그레스를 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 인그레스 이름을 입력하여 삭제한다.

4.2.8.3. 네트워크폴리시

네트워크폴리시는 네트워크 트래픽 제어를 위해 트래픽에 대한 규칙을 정의할 수 있다.

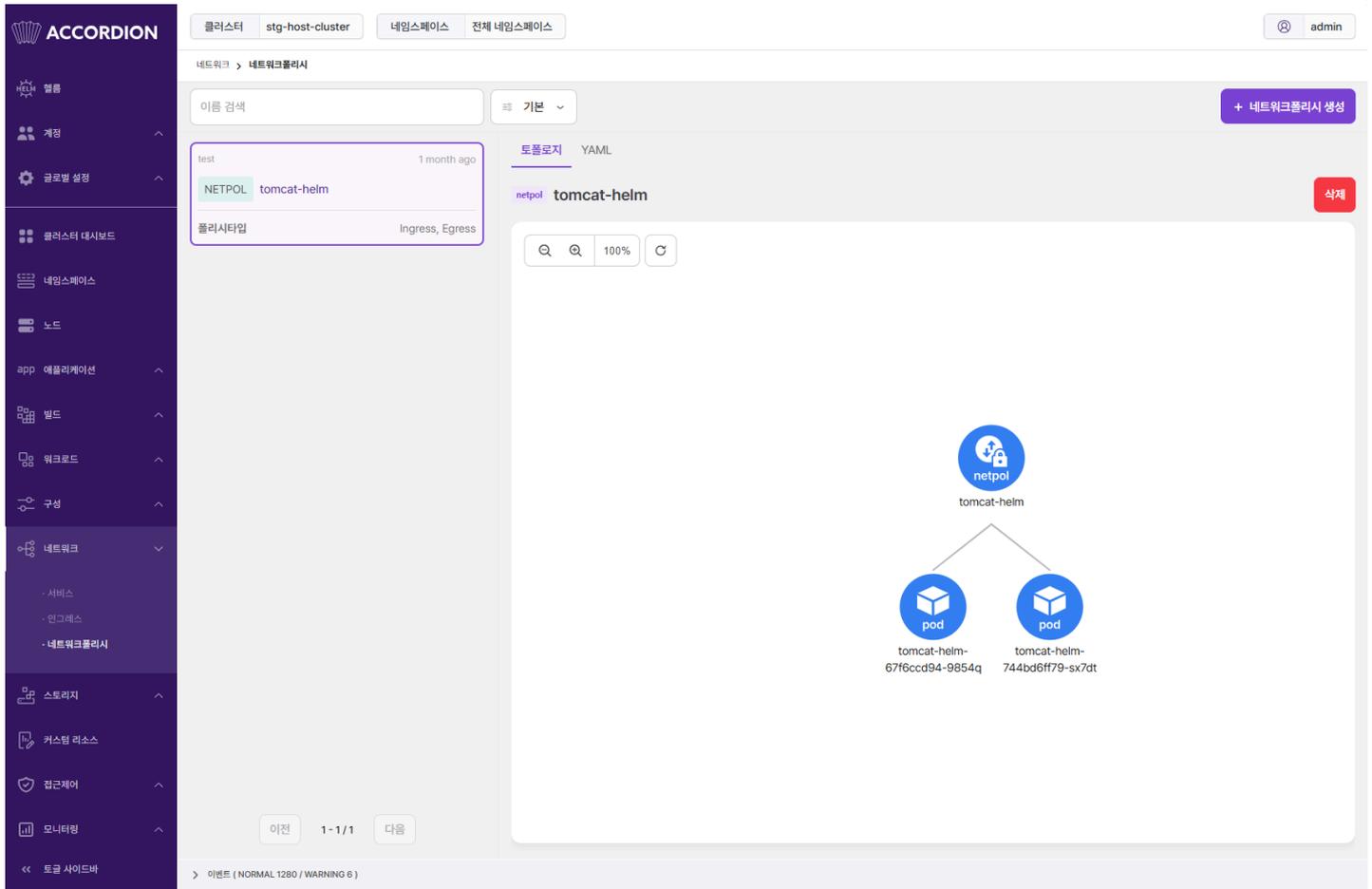
토폴로지

네트워크폴리시와 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

네트워크폴리시의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 네트워크폴리시
- 파드



네트워크폴리시 생성

+ 네트워크폴리시 생성 을 선택하면 나타나는 화면에서 쿠버네티스 네트워크폴리시 리소스 정보를 입력하여 생성할 수 있다.

네트워크 > 네트워크폴리시

← 네트워크폴리시 목록

+ 네트워크폴리시 생성

```

1  apiVersion: networking.k8s.io/v1
2  kind: NetworkPolicy
3  metadata:
4    name: example-networkpolicy
5    namespace: default
6  spec:
7    podSelector: {}
8    policyTypes: []
9

```

네트워크폴리시 수정

수정하려는 네트워크폴리시를 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

네트워크 > 네트워크폴리시

이름 검색

기본

+ 네트워크폴리시 생성

test 1 month ago

NETPOL tomcat-helm

폴리시타입 Ingress, Egress

토폴로지 YAML

netpol tomcat-helm

수정

삭제

```

1  kind: NetworkPolicy
2  apiVersion: networking.k8s.io/v1
3  metadata:
4    name: tomcat-helm
5    namespace: test
6    uid: 12d10405-3ecb-45a8-a940-f7436d2bed4e
7    resourceVersion: "13177923"
8    generation: 1
9    creationTimestamp: "2024-11-18T02:19:41Z"
10   labels:
11     app.kubernetes.io/instance: tomcat-helm
12     app.kubernetes.io/managed-by: Helm
13     app.kubernetes.io/name: tomcat
14     app.kubernetes.io/version: 10.1.33
15     helm.sh/chart: tomcat-11.3.1
16   annotations:
17     meta.helm.sh/release-name: tomcat-helm
18     meta.helm.sh/release-namespace: test
19   managedFields: ...

```

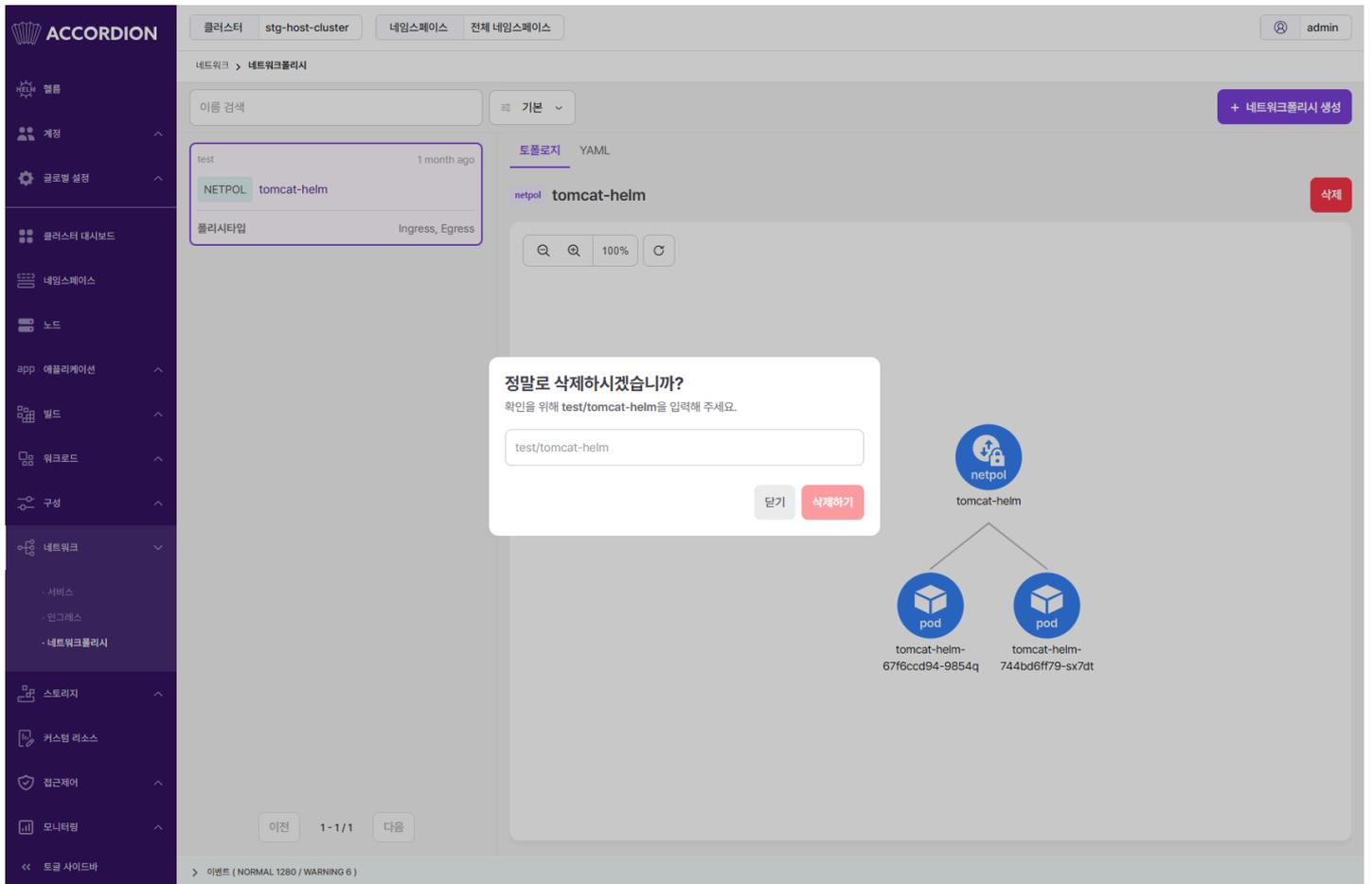
이전

1 - 1 / 1

다음

네트워크폴리시 삭제

삭제하려는 네트워크폴리시를 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 네트워크폴리시 이름을 입력하여 삭제한다.

4.2.9. 스토리지

컨테이너 환경에서 디스크는 일반 환경과 다른 특성을 가지고 있다. 컨테이너 내부에서 저장한 파일은 임시적인 것으로 컨테이너가 삭제되면 저장한 콘텐츠도 함께 삭제되며 파드의 집합으로 운영되는 경우를 고려하여 데이터의 공유에 대해서도 검토가 필요하다. 스토리지 리소스들을 이용해 이런 문제들을 해결할 수 있다. 스토리지 리소스들은 컨테이너가 삭제되더라도 데이터를 보존하고 파드의 집합에서 데이터를 공유하는 방법을 제공한다.

4.2.9.1. 퍼시스턴트볼륨

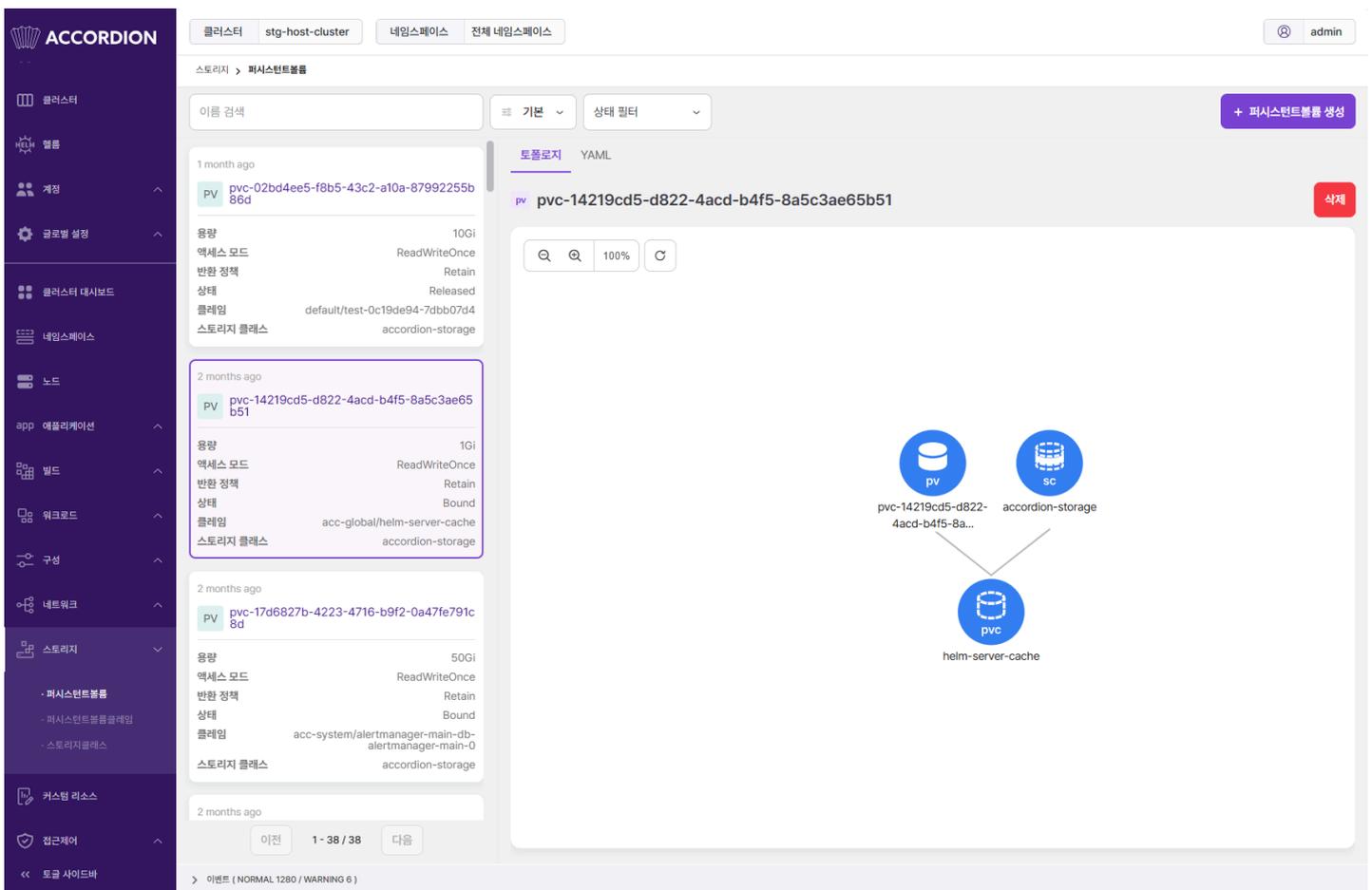
퍼시스턴트볼륨은 프로비저닝하거나 스토리지 클래스를 사용하여 동적으로 프로비저닝한 클러스터의 스토리지이다.

토폴로지

퍼시스턴트볼륨과 관련된 리소스를 토폴로지 형태로 보여준다. 기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

퍼시스턴트볼륨의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 퍼시스턴트볼륨, 스토리지클래스
- 퍼시스턴트볼륨클레임



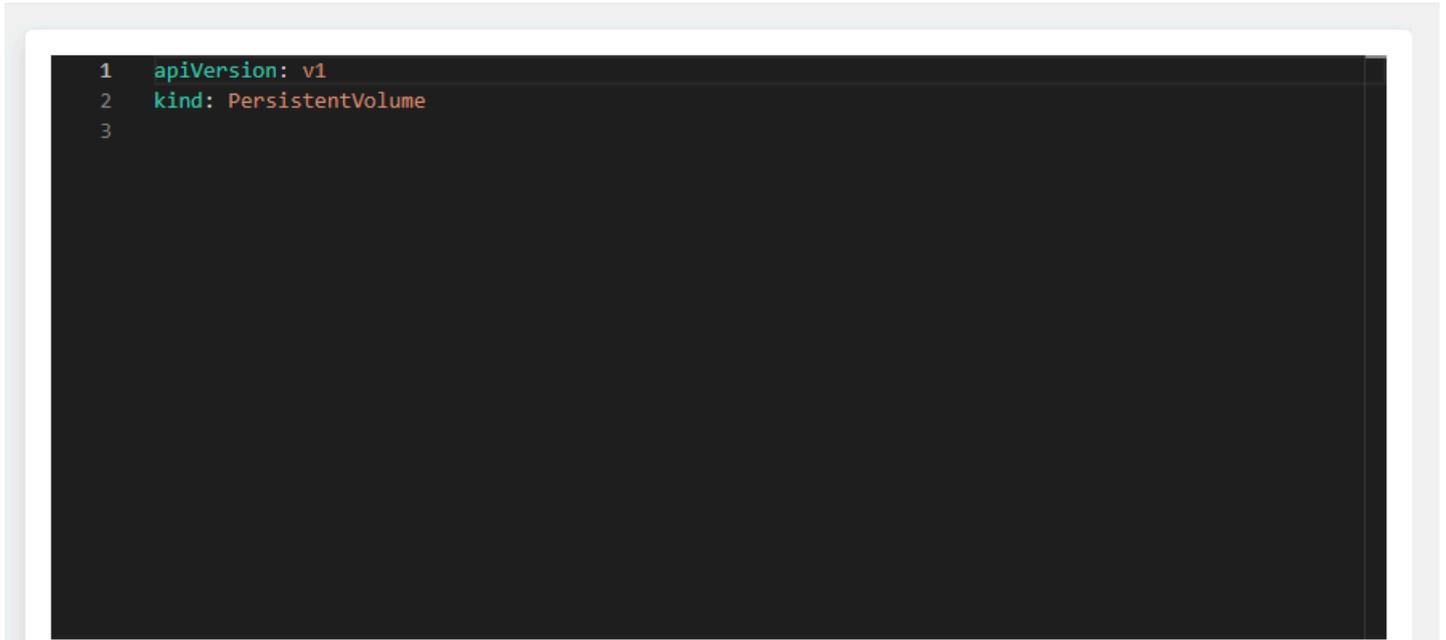
퍼시스턴트볼륨 생성

+ 퍼시스턴트볼륨 생성 을 선택하면 나타나는 화면에서 쿠버네티스 퍼시스턴트볼륨 리소스 정보를 입력하여 생성할 수 있다.

스토리지 > 퍼시스턴트볼륨

← 퍼시스턴트볼륨 목록

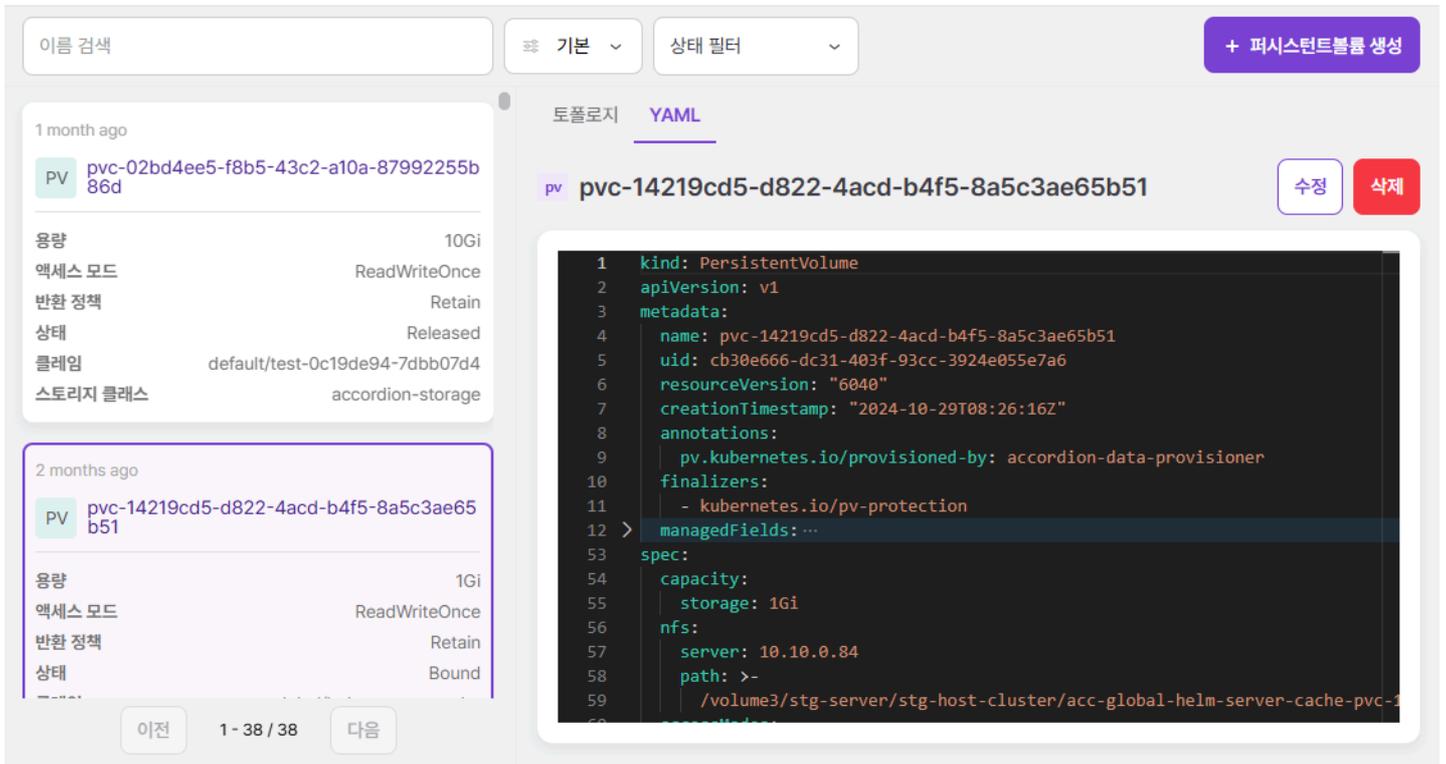
퍼시스턴트볼륨 생성



퍼시스턴트볼륨 수정

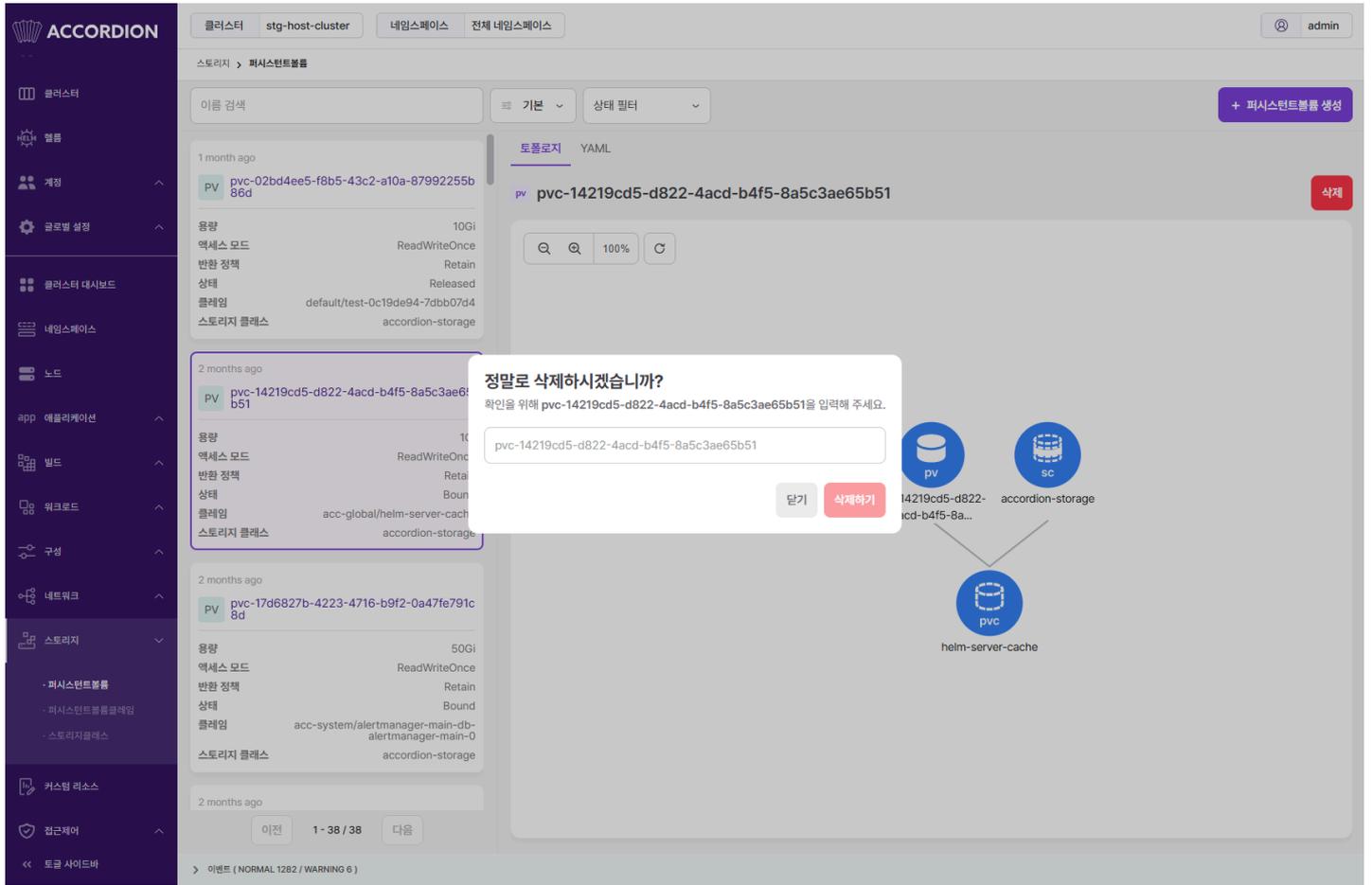
수정하려는 퍼시스턴트볼륨을 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

스토리지 > 퍼시스턴트볼륨



퍼시스턴트볼륨 삭제

삭제하려는 퍼시스턴트볼륨을 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 퍼시스턴트볼륨 이름을 입력하여 삭제한다.

4.2.9.2. 퍼시스턴트볼륨클레임

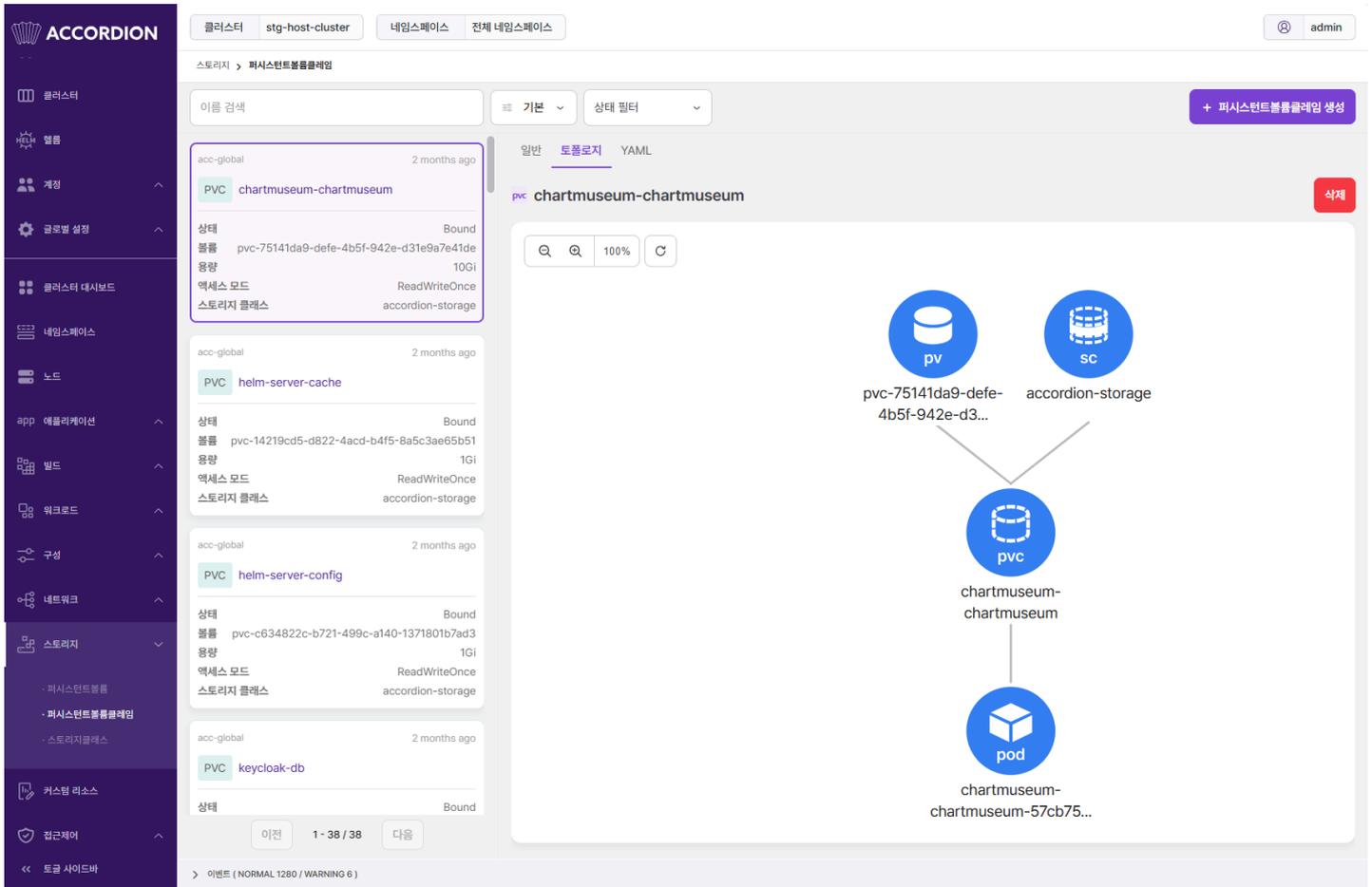
퍼시스턴트볼륨클레임은 사용자 스토리지에 대한 요청이다. 퍼시스턴트볼륨클레임은 퍼시스턴트볼륨 리소스를 사용하며 특정 크기 및 접근 모드를 요청할 수 있다.

토폴로지

퍼시스턴트볼륨클레임과 관련된 리소스를 토폴로지 형태로 보여준다. 기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

퍼시스턴트볼륨클레임의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 퍼시스턴트볼륨, 스토리지클래스
- 퍼시스턴트볼륨클레임
- 파드



퍼시스턴트볼륨클레임 생성

+ 퍼시스턴트볼륨클레임 생성 을 선택하면 나타나는 화면에서 쿠버네티스 퍼시스턴트볼륨클레임 리소스 정보를 입력하여 생성할 수 있다. 생성 시에는 FORM/YAML로 입력할 수 있다.

스토리지 > 퍼시스턴트볼륨클레임

← 퍼시스턴트볼륨클레임 목록

FORM YAML

퍼시스턴트볼륨클레임 생성

이름 *

네임스페이스 *

소스

스토리지 클래스 라벨 셀렉터

스토리지 클래스

셀렉터

	키		밸류

액세스 모드 *

ReadWriteOnce ReadOnlyMany ReadWriteMany ReadWriteOncePod

볼륨 모드 *

파일 시스템 블록

용량 *

 Gi

퍼시스턴트볼륨클레임 수정

수정하려는 퍼시스턴트볼륨클레임을 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

이름 검색

기본

상태 필터

+ 퍼시스턴트볼륨클레임 생성

acc-global 2 months ago

PVC chartmuseum-chartmuseum

상태 Bound

볼륨 pvc-75141da9-defe-4b5f-942e-d31e9a7e41de

용량 10Gi

액세스 모드 ReadWriteOnce

스토리지 클래스 accordion-storage

acc-global 2 months ago

PVC helm-server-cache

상태 Bound

볼륨 pvc-14219cd5-d822-4acd-b4f5-8a5c3ae65b51

용량 10Gi

액세스 모드 ReadWriteOnce

스토리지 클래스 accordion-storage

이전 1 - 38 / 38 다음

일반 토폴로지 YAML

pvc chartmuseum-chartmuseum

수정 삭제

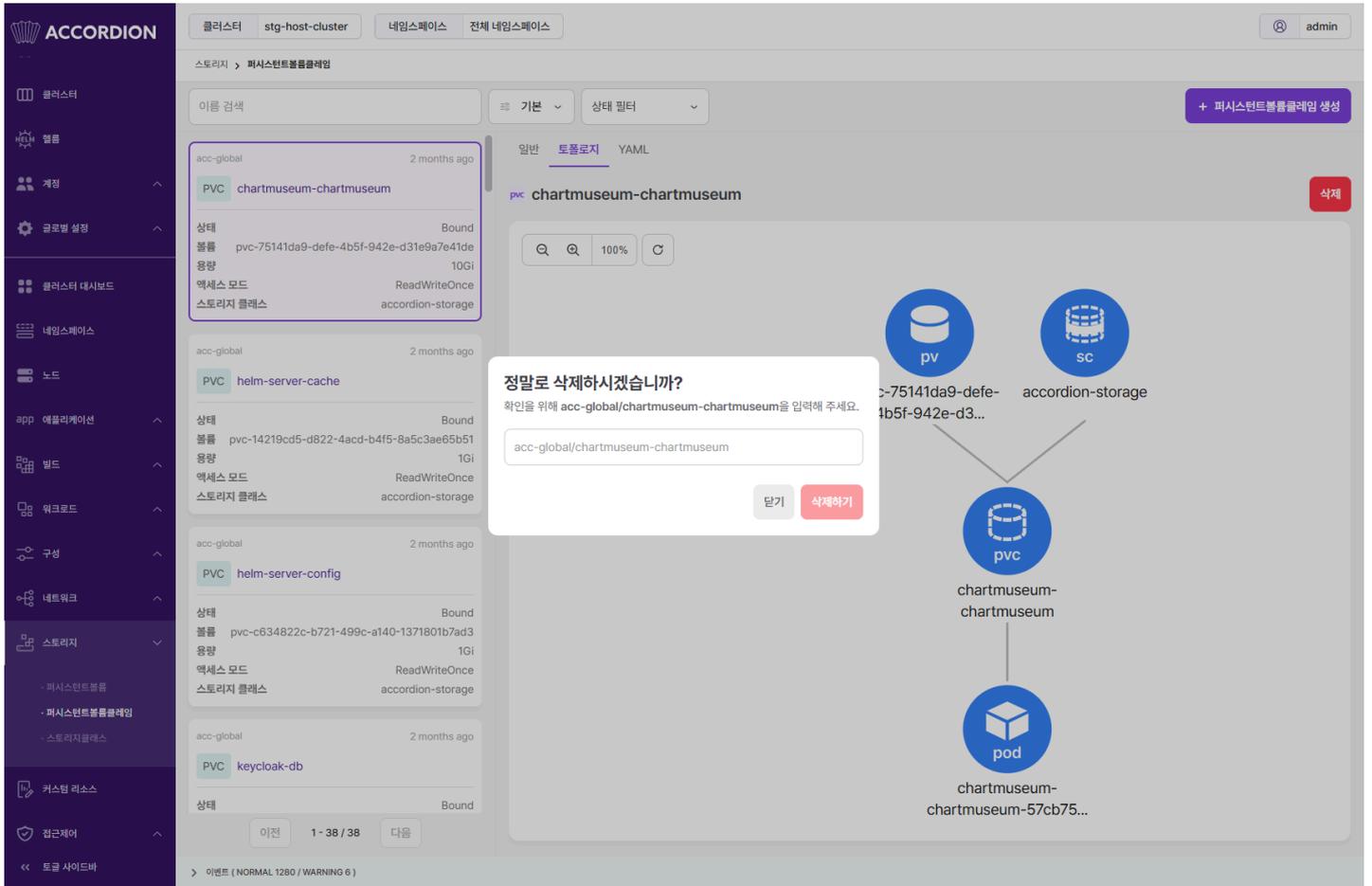
```

1 kind: PersistentVolumeClaim
2 apiVersion: v1
3 metadata:
4   name: chartmuseum-chartmuseum
5   namespace: acc-global
6   uid: 75141da9-defe-4b5f-942e-d31e9a7e41de
7   resourceVersion: "4888"
8   creationTimestamp: "2024-10-29T08:25:19Z"
9   labels:
10    app: chartmuseum-chartmuseum
11    release: chartmuseum
12  annotations:
13    kubectl.kubernetes.io/last-applied-configuration: >
14      {"apiVersion":"v1","kind":"PersistentVolumeClaim","metadata":{"annotat
15    pv.kubernetes.io/bind-completed:"yes"
16    pv.kubernetes.io/bound-by-controller:"yes"
17    release-name: chartmuseum
18    release-namespace: acc-global
19    volume.beta.kubernetes.io/storage-provisioner: accordion-data-provisioner

```

퍼시스턴트볼륨클레임 삭제

삭제하려는 퍼시스턴트볼륨클레임을 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 퍼시스턴트볼륨클레임 이름을 입력하여 삭제한다.

4.2.9.3. 스토리지클래스

스토리지클래스는 해당 스토리지클래스에 속하는 퍼시스턴트볼륨을 동적으로 프로비저닝한다.

스토리지 > 스토리지클래스

이름 검색 기본 + 스토리지클래스 생성

1 week ago

SC accordion-storage

프로비저너 accordion-data-provisioner
반환 정책 Retain

이전 1 - 1 / 1 다음

YAML

sc accordion-storage 수정 삭제

```

1 kind: StorageClass
2 apiVersion: storage.k8s.io/v1
3 metadata:
4   name: accordion-storage
5   uid: dca418b9-2469-4e09-a281-c926bf51f4ce
6   resourceVersion: "36769259"
7   creationTimestamp: "2024-08-21T03:32:49Z"
8   labels:
9     accordions.co.kr/display: hidden
10    accordions.co.kr/kind: system
11   annotations:
12     kubectl.kubernetes.io/last-applied-configuration: >
13     {"apiVersion":"storage.k8s.io/v1","kind":"StorageClass","metadata":{"annotati
14 > managedFields: ...
36 provisioner: accordion-data-provisioner
37 parameters:
38   archiveOnDelete: "false"
39 reclaimPolicy: Retain
40 mountOptions:
41   - nfsvers=3,noLOCK

```

스토리지클래스 생성

+ 스토리지클래스 생성 을 선택하면 나타나는 화면에서 쿠버네티스 스토리지클래스 리소스 정보를 입력하여 생성할 수 있다.

스토리지 > 스토리지클래스

← 스토리지클래스 목록 스토리지클래스 생성

```

1 apiVersion: storage.k8s.io/v1
2 kind: StorageClass
3 metadata:
4   name: example-storageclass
5 provisioner: example-provisioner
6 reclaimPolicy: Retain
7

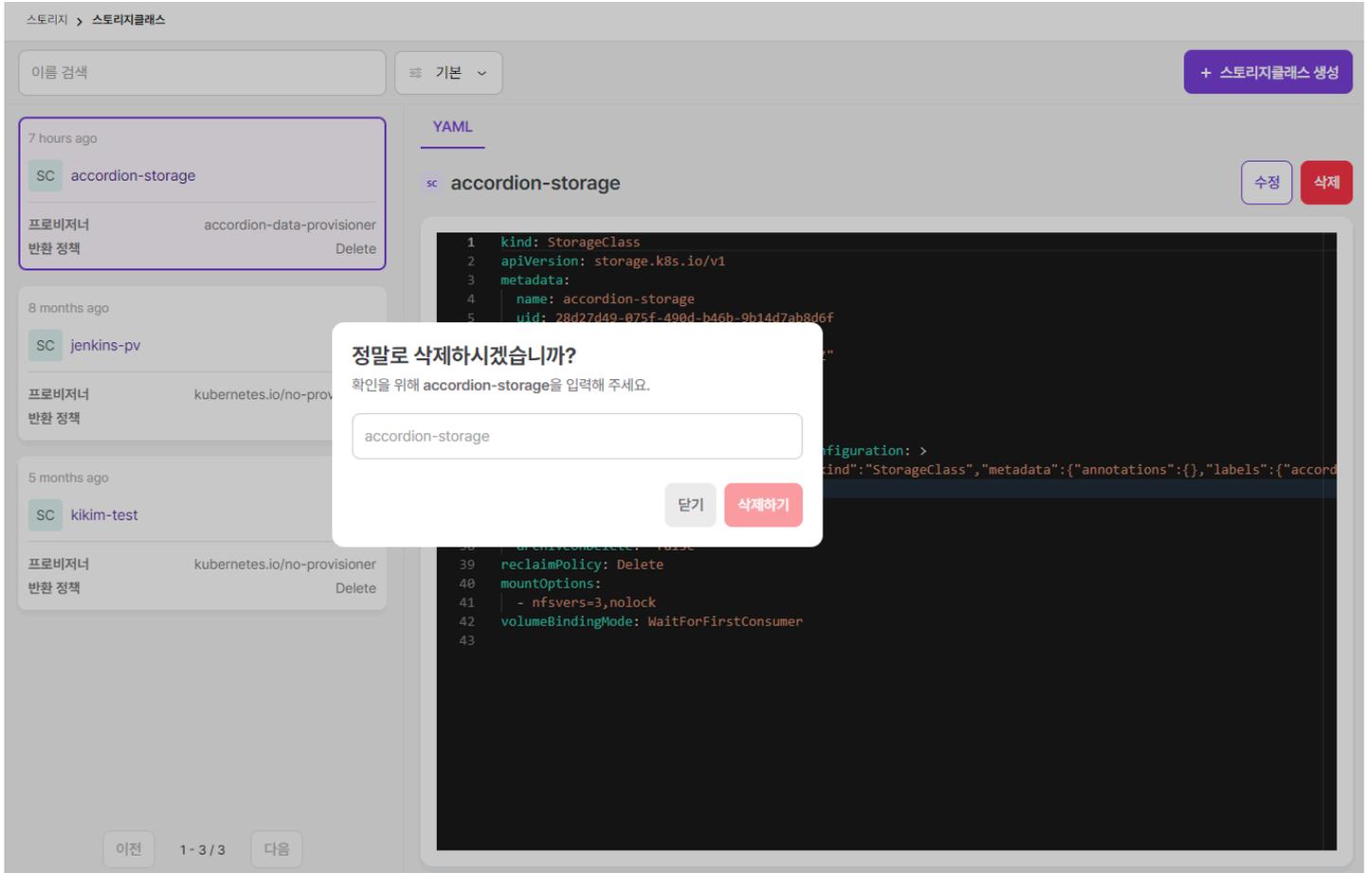
```

스토리지클래스 수정

수정하려는 스토리지클래스를 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

스토리지클래스 삭제

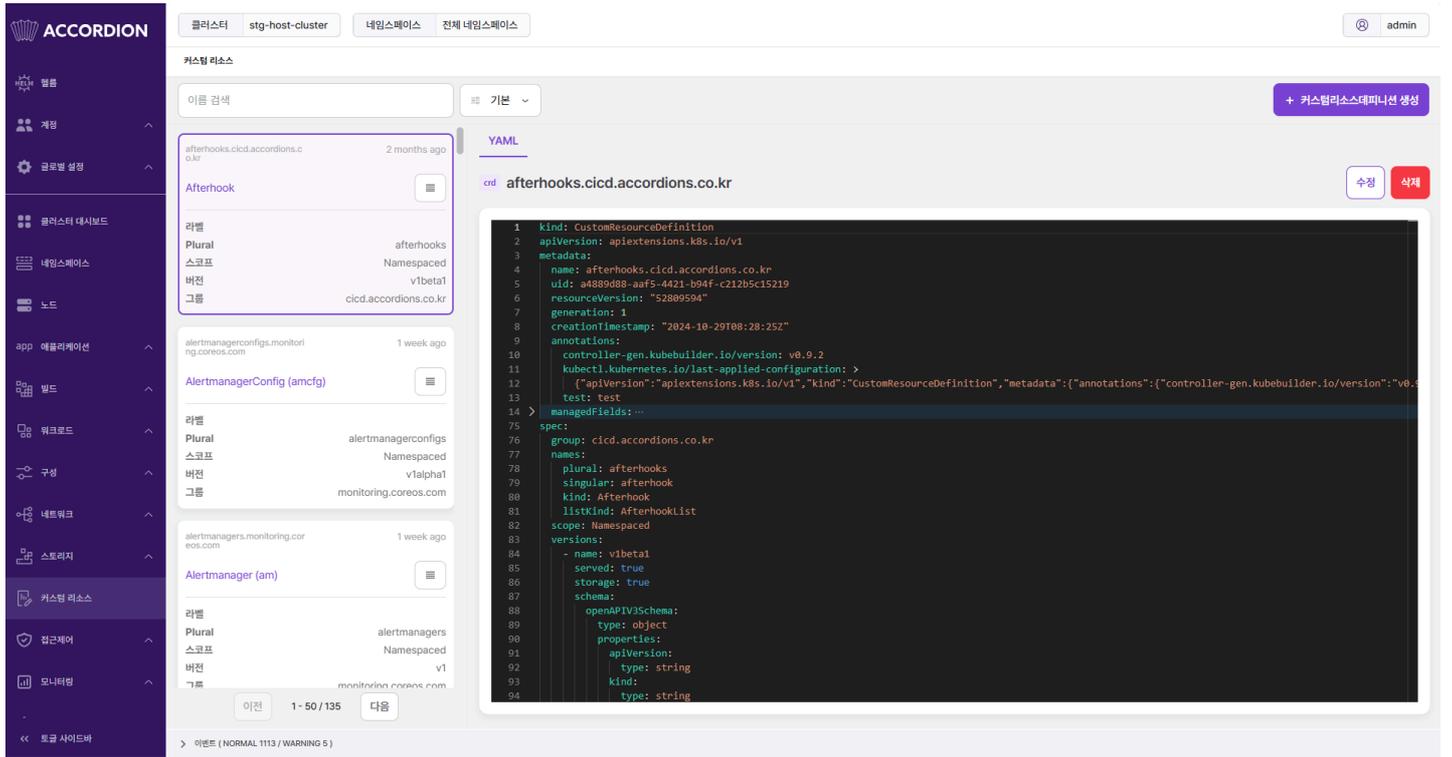
삭제하려는 스토리지클래스를 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 스토리지클래스 이름을 입력하여 삭제한다.

4.2.10. 커스텀 리소스

커스텀 리소스는 사용자가 정의한 오브젝트를 생성할 수 있다. 쿠버네티스에서 제공하는 리소스 외에도 필요한 리소스를 쿠버네티스 내부에 사용할 수 있다.



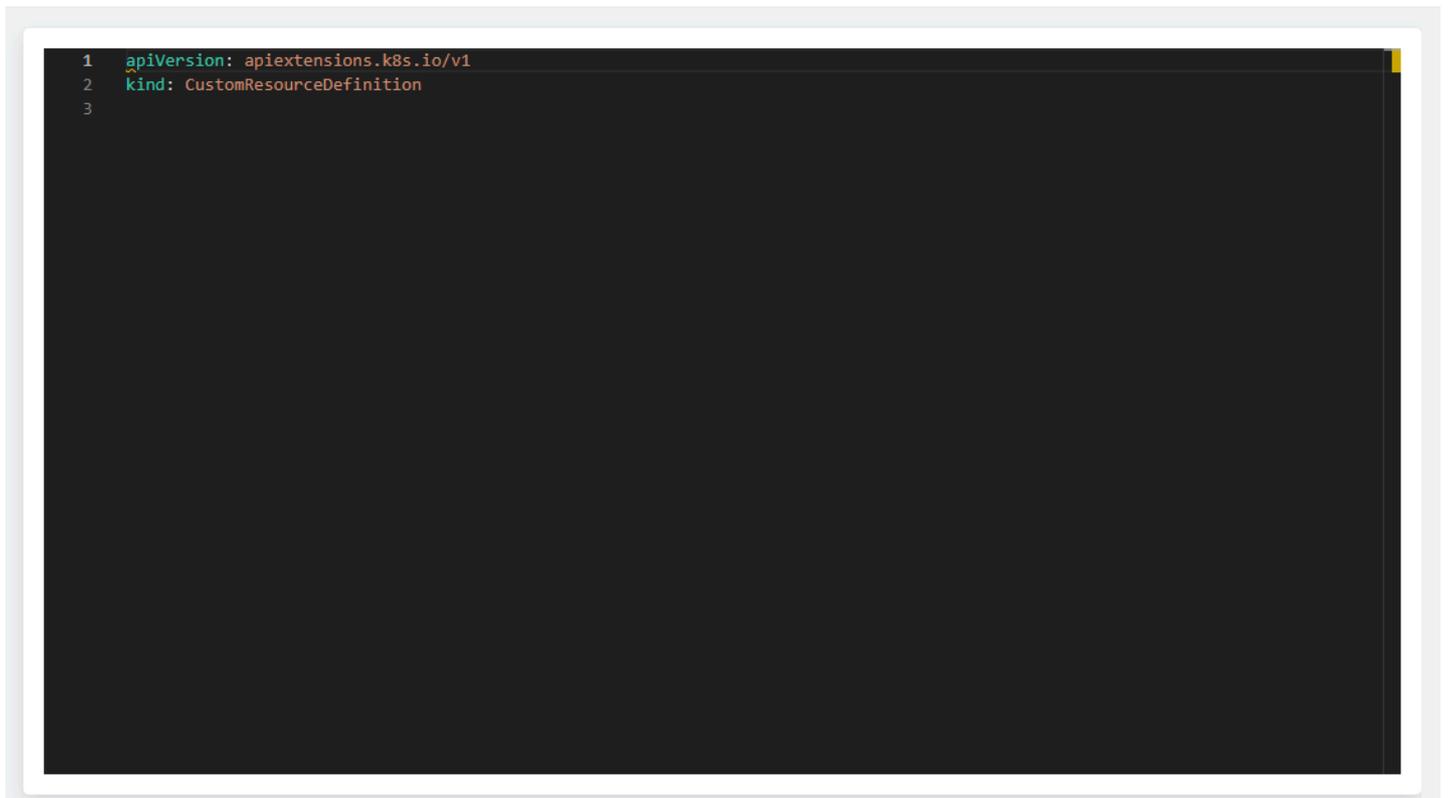
4.2.10.1. 커스텀 리소스 생성

+ 커스텀리소스데피니션 생성 을 선택하면 나타나는 화면에서 쿠버네티스 커스텀 리소스 리소스 정보를 입력하여 생성할 수 있다.

커스텀 리소스

← 커스텀리소스데피니션 목록

커스텀리소스데피니션 생성

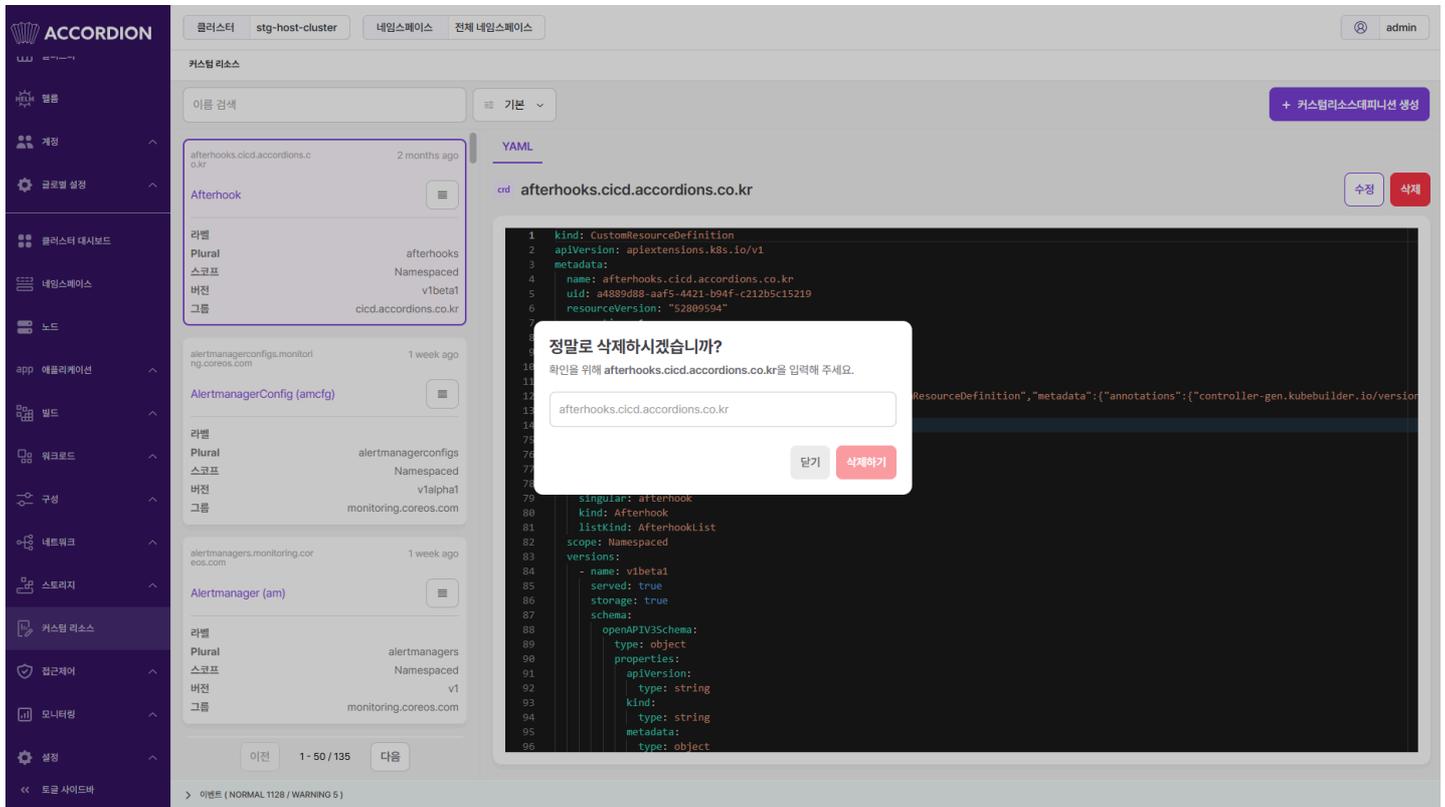


4.2.10.2. 커스텀 리소스 수정

수정하려는 커스텀 리소스를 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

4.2.10.3. 커스텀 리소스 삭제

삭제하려는 커스텀 리소스를 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 커스텀 리소스 이름을 입력하여 삭제한다.

4.2.11. 접근제어

접근제어는 쿠버네티스의 RBAC(역할 기반 접근 제어)를 기반으로 리소스에 대한 사용자의 접근을 제어한다.

4.2.11.1. 룰

룰은 특정 네임스페이스의 API나 리소스에 대한 권한을 명시한다.

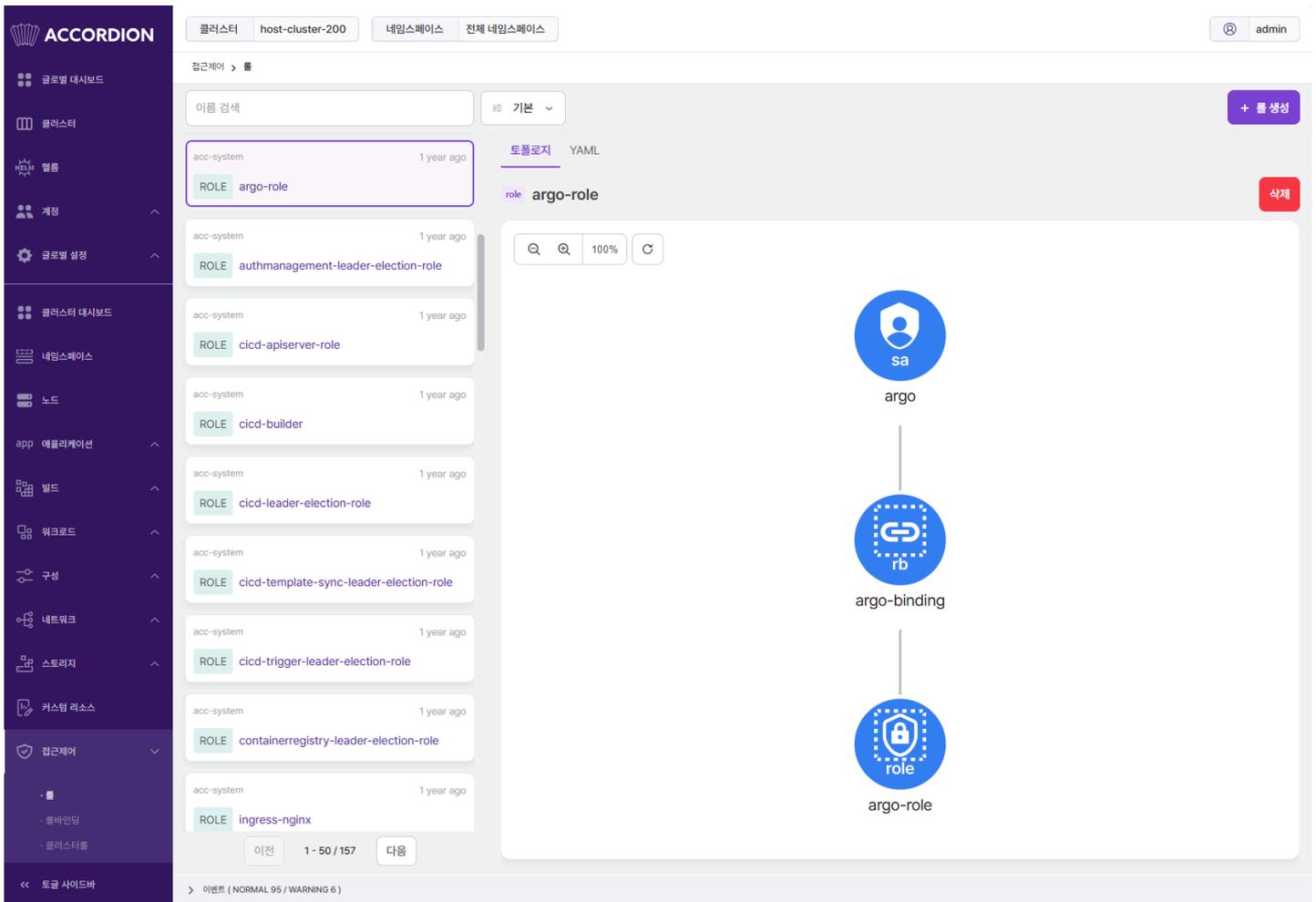
토폴로지

룰과 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

룰의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 서비스어카운트
- 룰바인딩
- 룰



롤 생성

+ 롤 생성 을 선택하면 나타나는 화면에서 쿠버네티스 롤 리소스 정보를 입력하여 생성할 수 있다.

접근제어 > 롤

← 롤 목록

+ 롤 생성

```

1  apiVersion: rbac.authorization.k8s.io/v1
2  kind: Role
3  metadata:
4    name: example-role
5    namespace: default
6  rules:
7    - apiGroups:
8      - ""
9      resources:
10     - pods
11     verbs:
12     - get
13     - watch
14     - list
15

```

롤 수정

수정하려는 롤을 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

접근제어 > 롤

이름 검색

기본

+ 롤 생성

acc-system
1 year ago

ROLE

argo-role

acc-system
1 year ago

ROLE

authmanagement-leader-election-role

acc-system
1 year ago

ROLE

cicd-apiserver-role

acc-system
1 year ago

ROLE

cicd-builder

acc-system
1 year ago

ROLE

토폴로지
YAML

role argo-role

수정

삭제

```

1  kind: Role
2  apiVersion: rbac.authorization.k8s.io/v1
3  metadata:
4    name: argo-role
5    namespace: acc-system
6    uid: ac09a8a7-58a4-41e6-a08b-7242647e2bb1
7    resourceVersion: '6366'
8    creationTimestamp: '2023-11-02T01:54:39Z'
9    annotations:
10     kubectl.kubernetes.io/last-applied-configuration: >
11     {"apiVersion":"rbac.authorization.k8s.io/v1","kind":"Role","metadata":{"
12 > managedFields: ...
24  rules:
25    - verbs:
26     - create
27     - get
28     - update
29    apiGroups:
30     - coordination.k8s.io

```

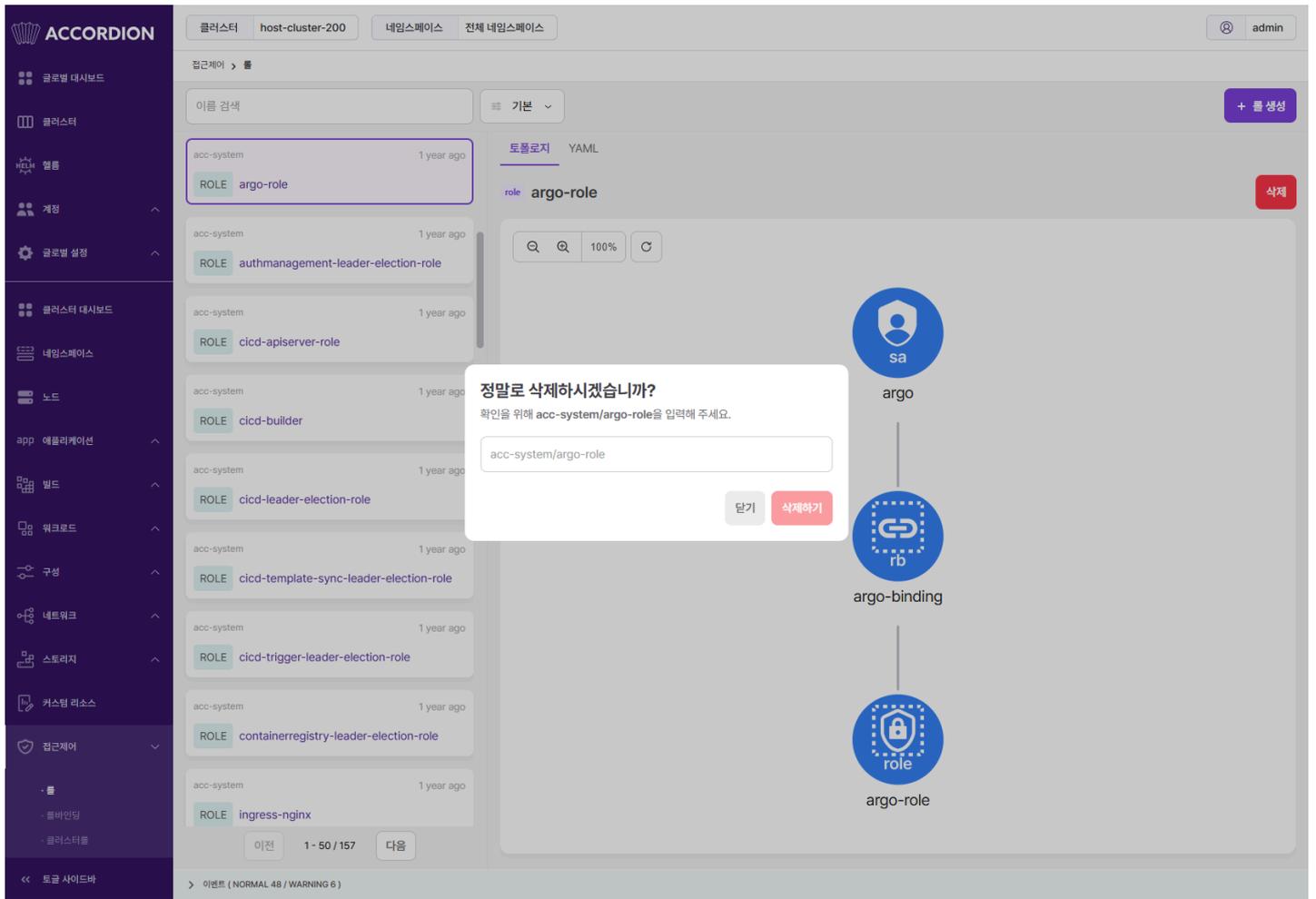
이전

1 - 50 / 157

다음

롤 삭제

삭제하려는 롤을 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 롤 이름을 입력하여 삭제한다.

4.2.11.2. 롤바인딩

롤바인딩은 특정 네임스페이스에 롤/클러스터롤과 서비스어카운트를 연결해주며 지정된 서비스어카운트들이 명시된 롤을 사용할 수 있도록 한다.

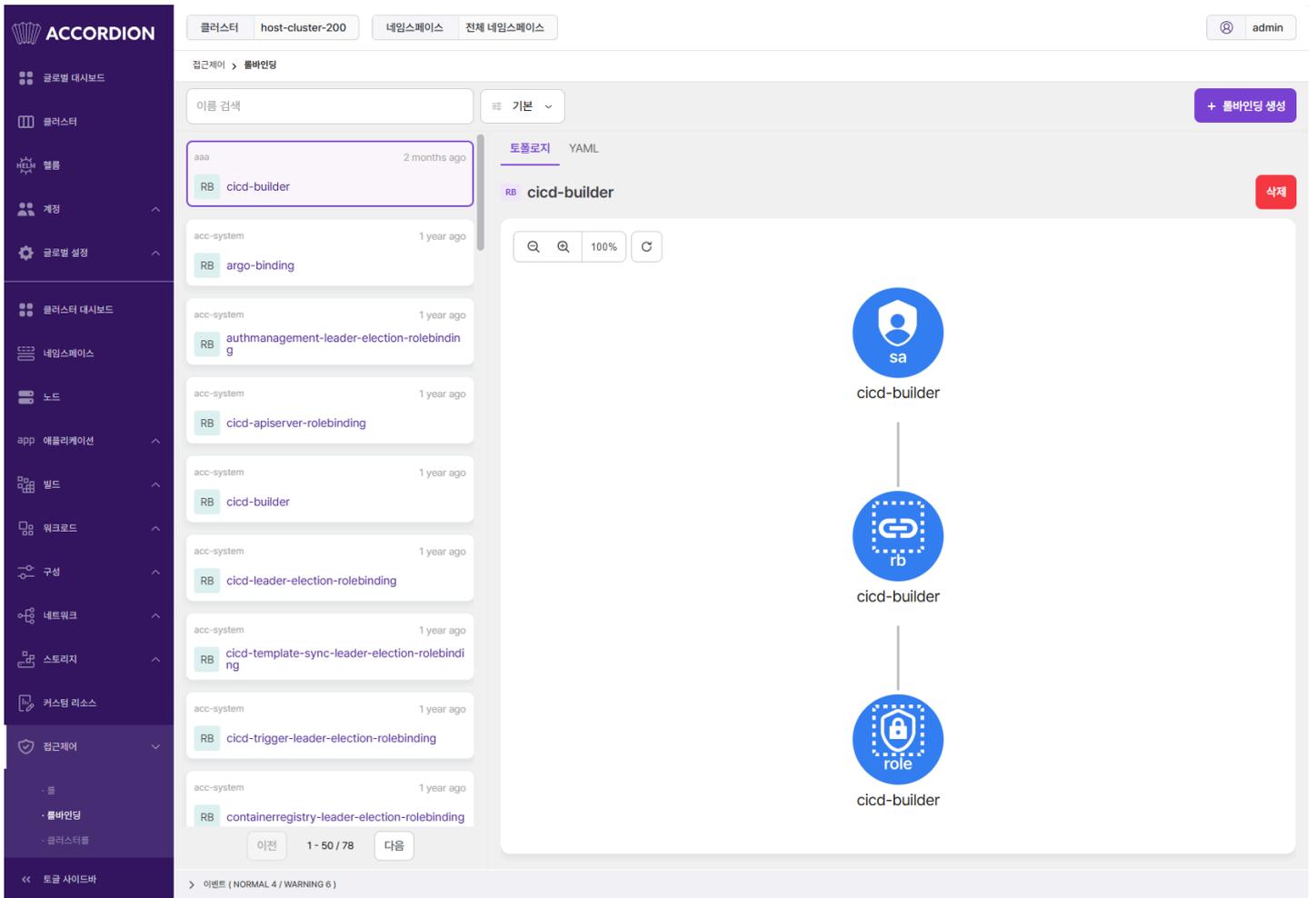
토폴로지

롤바인딩과 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

롤바인딩의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 서비스어카운트
- 롤바인딩
- 롤



롤바인딩 생성

+ 롤바인딩 생성 을 선택하면 나타나는 화면에서 쿠버네티스 롤바인딩 리소스 정보를 입력하여 생성할 수 있다.

접근제어 > 롤바인딩

← 롤바인딩 목록

롤바인딩 생성

```

1  apiVersion: rbac.authorization.k8s.io/v1
2  kind: RoleBinding
3  metadata:
4    name: example-role-binding
5    namespace: default
6  subjects:
7    - kind: ServiceAccount
8      name: example-serviceaccount
9      namespace: default
10 roleRef:
11   kind: Role
12   name: example-role
13   apiGroup: rbac.authorization.k8s.io
14

```

롤바인딩 수정

수정하려는 롤바인딩을 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

접근제어 > 롤바인딩

이름 검색

기본

+ 롤바인딩 생성

- aaa 2 months ago
RB cicd-builder
- acc-system 1 year ago
RB argo-binding
- acc-system 1 year ago
RB authmanagement-leader-election-rolebinding
- acc-system 1 year ago
RB cicd-apiserver-rolebinding
- acc-system 1 year ago

이전

1 - 50 / 78

다음

토폴로지 YAML

RB cicd-builder

수정

삭제

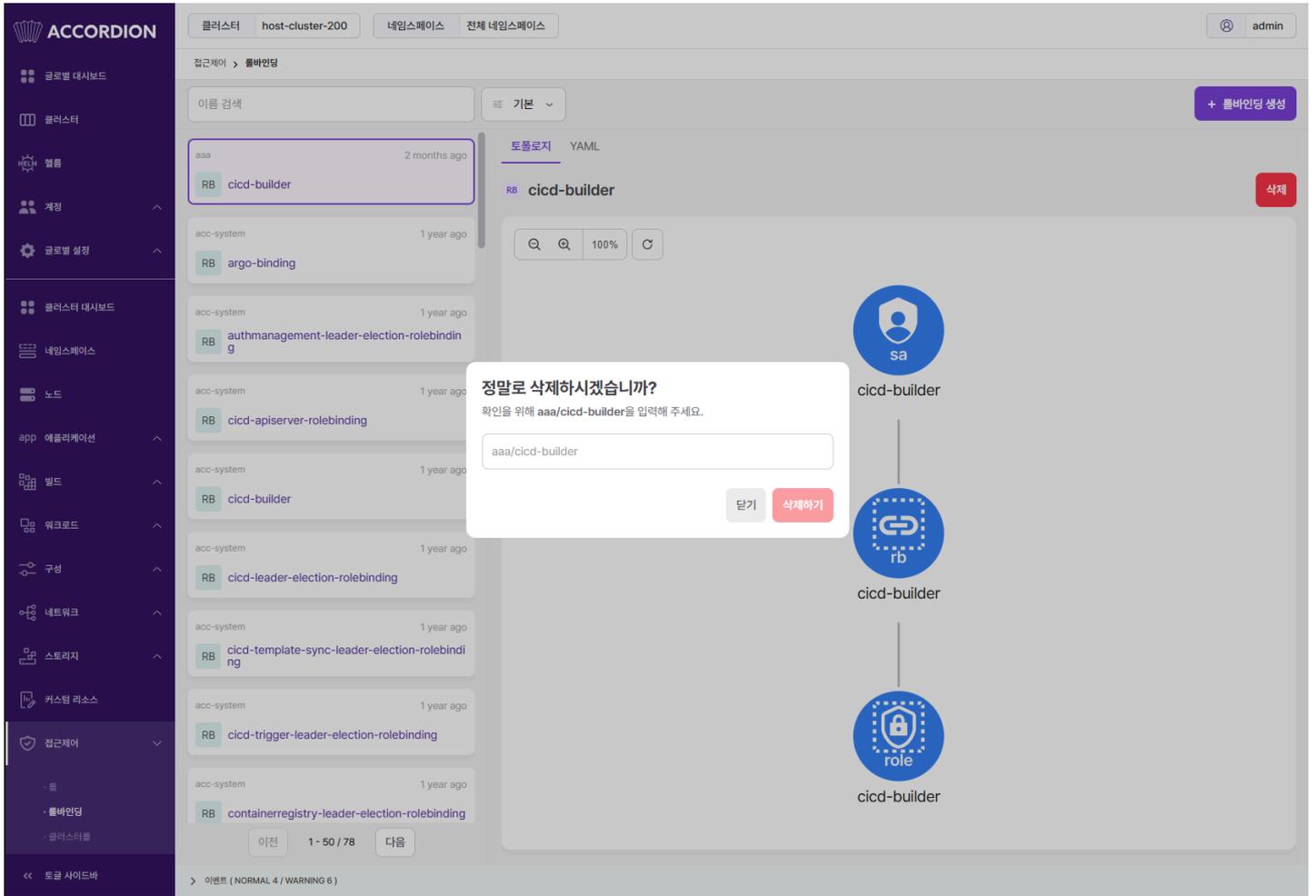
```

1  kind: RoleBinding
2  apiVersion: rbac.authorization.k8s.io/v1
3  metadata:
4    name: cicd-builder
5    namespace: aaa
6    uid: aebde98c-f5e2-4038-a230-4438afd13c4e
7    resourceVersion: "255256460"
8    creationTimestamp: "2024-10-23T07:04:23Z"
9    labels:
10   app.kubernetes.io/created-by: cicd-manager
11   app.kubernetes.io/managed-by: cicd-manager
12  managedFields: ...
26 subjects:
27   - kind: ServiceAccount
28     name: cicd-builder
29 roleRef:
30   apiGroup: rbac.authorization.k8s.io
31   kind: Role
32   name: cicd-builder

```

롤바인딩 삭제

삭제하려는 롤바인딩을 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 롤바인딩 이름을 입력하여 삭제한다.

4.2.11.3. 클러스터롤

클러스터롤은 클러스터의 API나 리소스에 대한 권한을 명시한다.

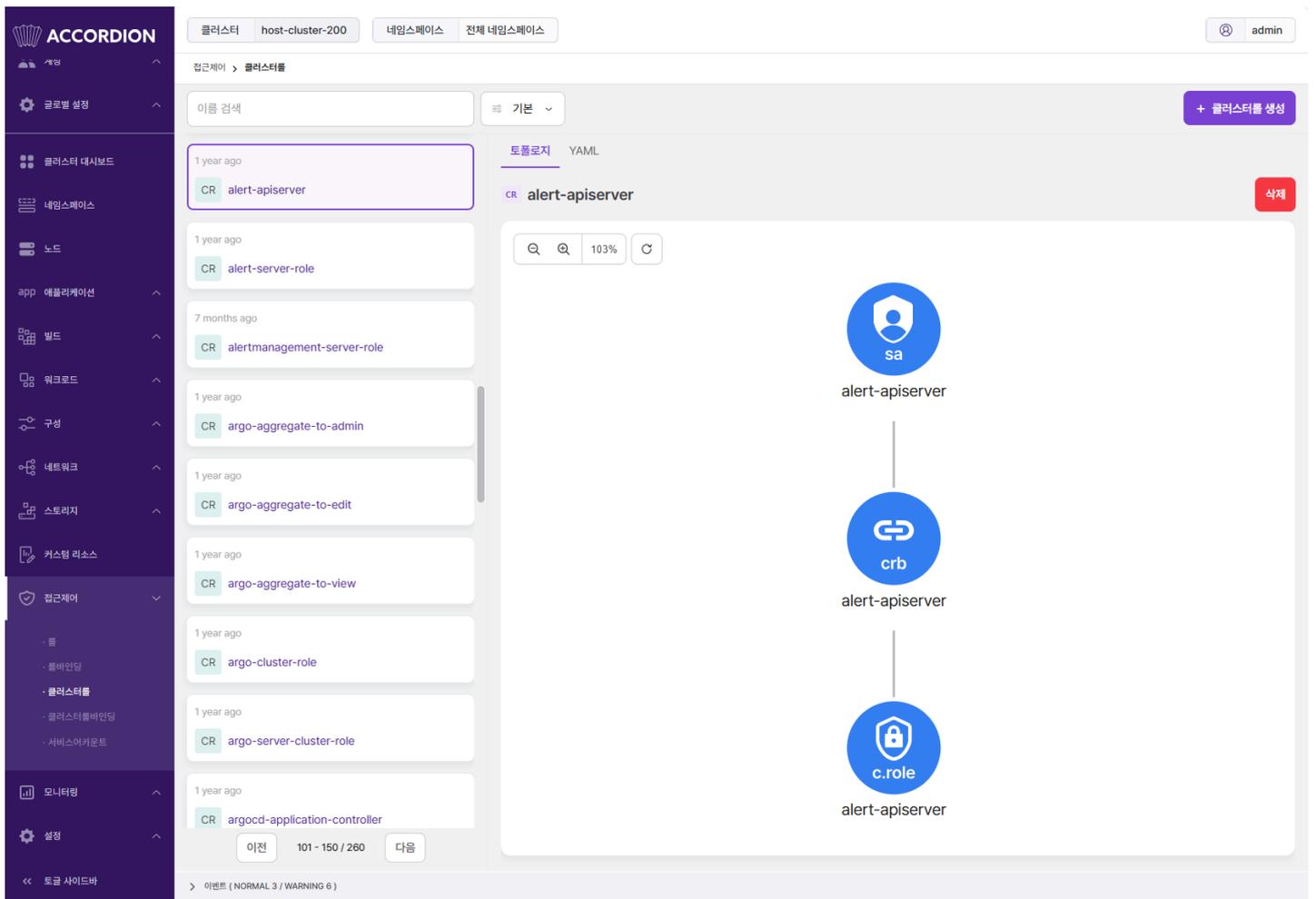
토폴로지

클러스터롤과 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

클러스터롤의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 서비스어카운트
- 클러스터롤바인딩
- 클러스터롤



클러스터를 생성

+ 클러스터를 생성 을 선택하면 나타나는 화면에서 쿠버네티스 클러스터를 리소스 정보를 입력하여 생성할 수 있다.

접근제어 > 클러스터들

← 클러스터를 목록

클러스터를 생성

```

1  apiVersion: rbac.authorization.k8s.io/v1
2  kind: ClusterRole
3  metadata:
4  |   name: ''

```

클러스터를 수정

수정하려는 클러스터를 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

접근제어 > 클러스터들

이름 검색

🔍 기본

+ 클러스터를 생성

1 year ago
CR alert-apiserver

1 year ago
CR alert-server-role

7 months ago
CR alertmanagement-server-role

1 year ago
CR argo-aggregate-to-admin

1 year ago

이전
101 - 150 / 260
다음

토폴로지
YAML

CR alert-apiserver
 수정
삭제

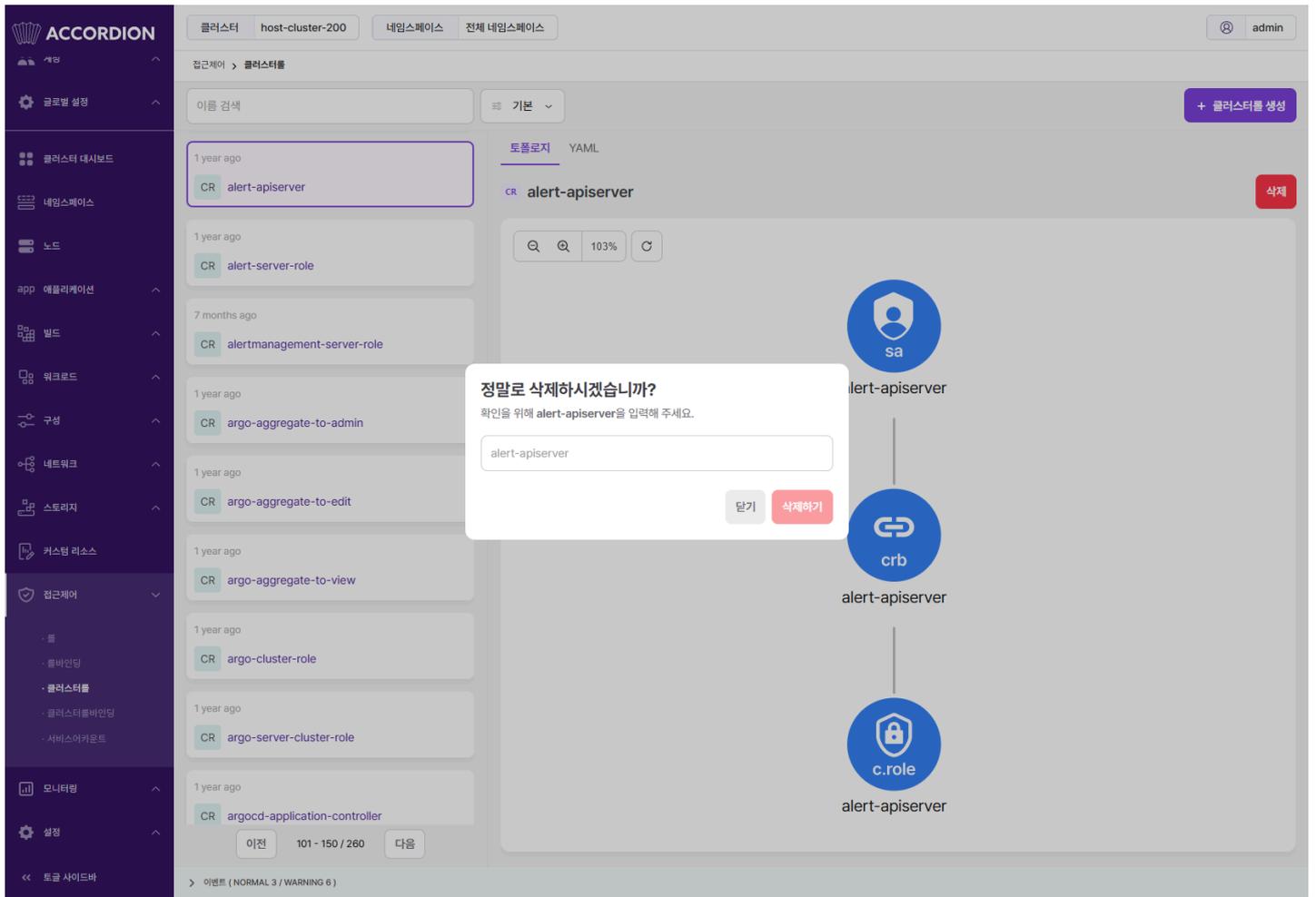
```

1  kind: ClusterRole
2  apiVersion: rbac.authorization.k8s.io/v1
3  metadata:
4  |   name: alert-apiserver
5  |   uid: 3fbec954-5bac-464d-96e4-216ccb2138c6
6  |   resourceVersion: "9335"
7  |   creationTimestamp: "2023-11-02T01:59:01Z"
8  |   annotations:
9  |     |   kubect1.kubernetes.io/last-applied-configuration: >
10 |     |     {"apiVersion":"rbac.authorization.k8s.io/v1","kind":"ClusterRole","meta
11 > |     |     managedFields: ...
23  rules:
24  |   - verbs:
25  |     |   - get
26  |     |   - list
27  |     |   - watch
28  |     apiGroups:
29  |       |   - management.accordions.co.kr
30  |       resourceNames:

```

클러스터를 삭제

삭제하려는 클러스터를 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 클러스터 이름을 입력하여 삭제한다.

4.2.11.4. 클러스터롤바인딩

클러스터롤바인딩은 클러스터롤과 서비스어카운트를 연결해주며 지정된 서비스어카운트들이 명시된 클러스터롤을 사용할 수 있도록 한다.

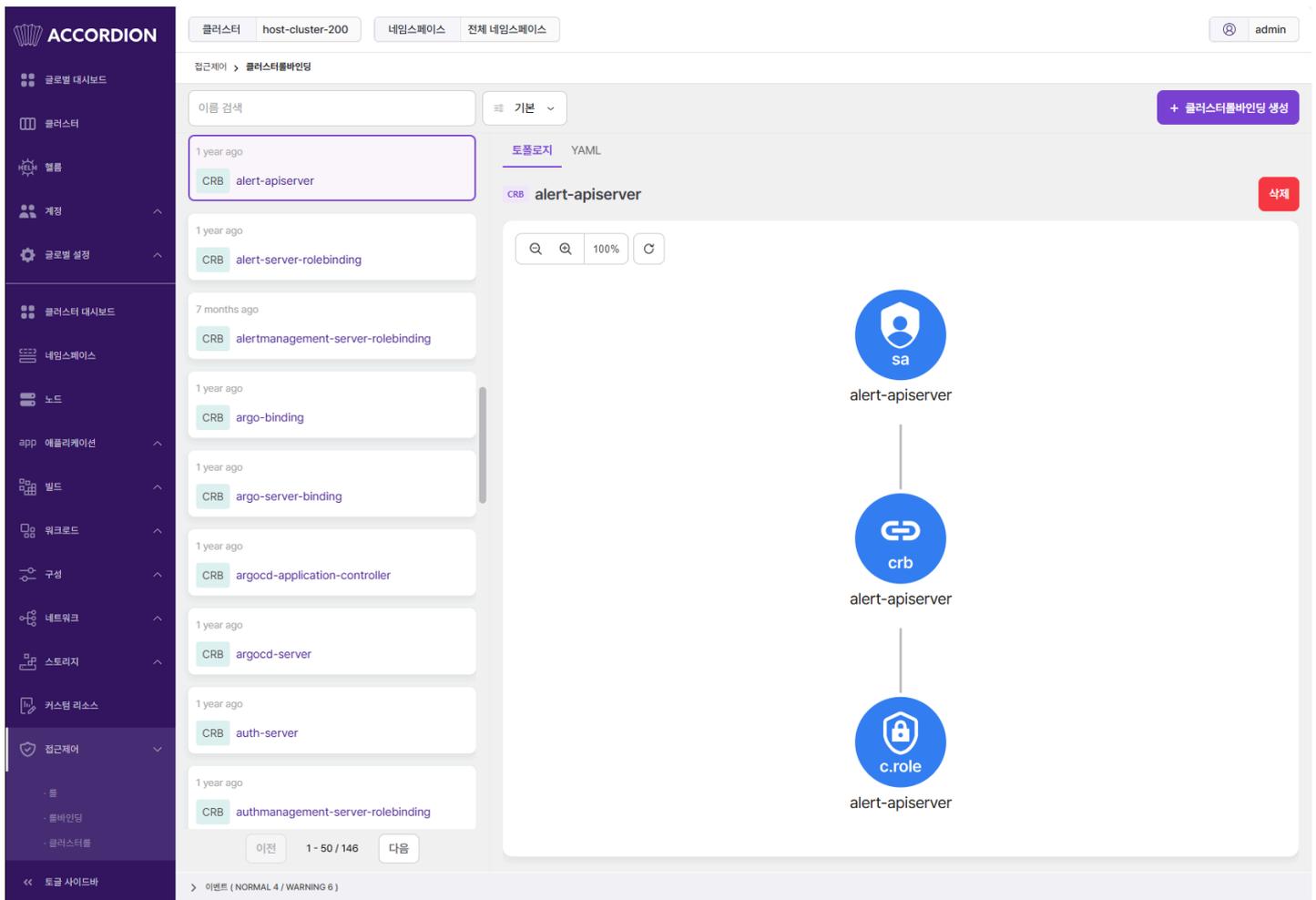
토폴로지

클러스터롤바인딩과 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

클러스터롤바인딩의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 서비스어카운트
- 클러스터롤바인딩
- 클러스터롤



클러스터롤바인딩 생성

+ 클러스터롤바인딩 생성 을 선택하면 나타나는 화면에서 쿠버네티스 클러스터롤바인딩 리소스 정보를 입력하여 생성할 수 있다.

접근제어 > 클러스터롤바인딩

← 클러스터롤바인딩 목록

클러스터롤바인딩 생성

```

1 apiVersion: rbac.authorization.k8s.io/v1
2 kind: ClusterRoleBinding
3 metadata:
4   name: ''

```

클러스터롤바인딩 수정

수정하려는 클러스터롤바인딩을 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

접근제어 > 클러스터롤바인딩

이름 검색

기본

+ 클러스터롤바인딩 생성

1 year ago

CRB alert-apiserver

1 year ago

CRB alert-server-rolebinding

7 months ago

CRB alertmanagement-server-rolebinding

1 year ago

CRB argo-binding

1 year ago

이전

1 - 50 / 146

다음

도플로지 YAML

CRB alert-apiserver

수정

삭제

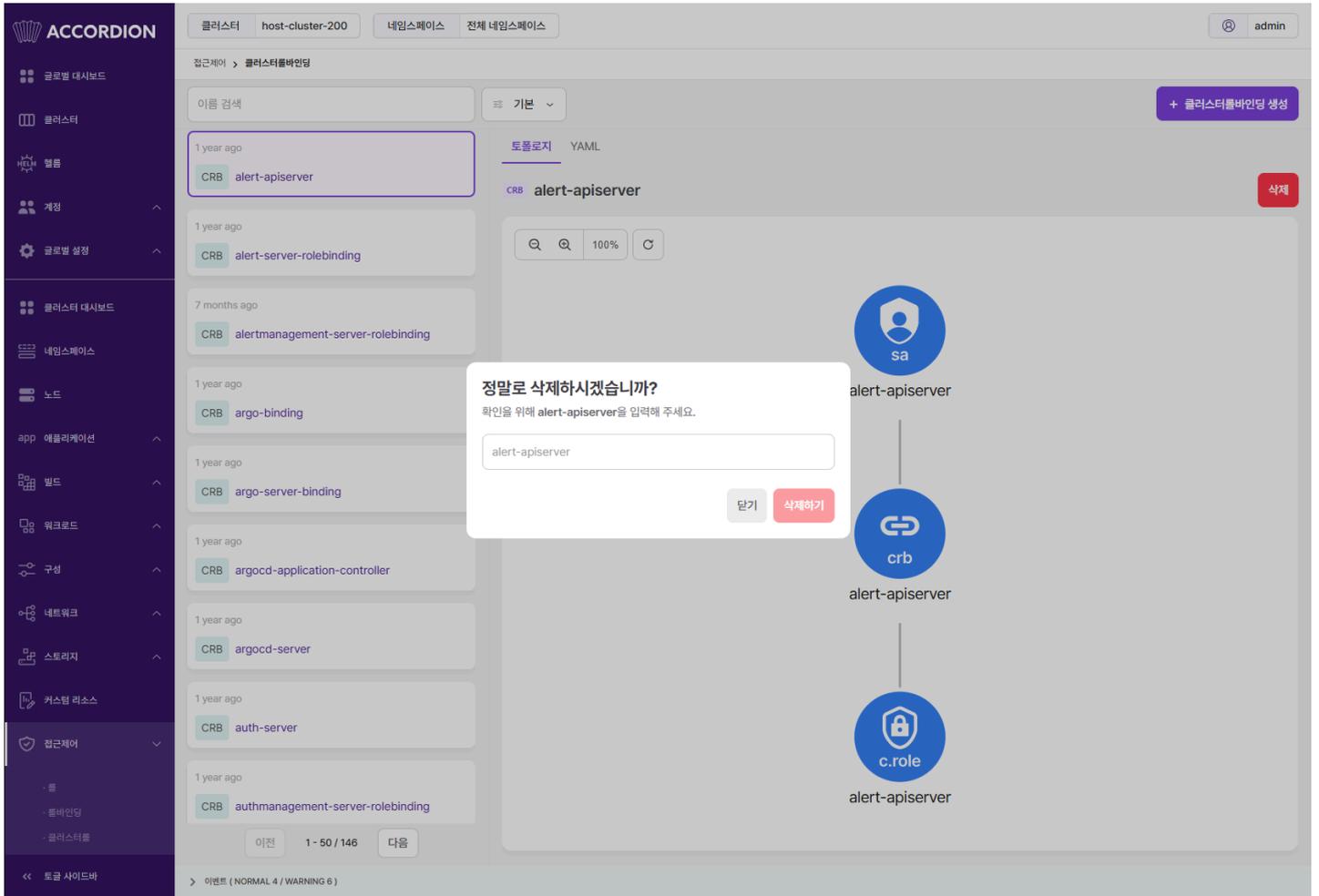
```

1 kind: ClusterRoleBinding
2 apiVersion: rbac.authorization.k8s.io/v1
3 metadata:
4   name: alert-apiserver
5   uid: af247de0-35e3-419e-a253-f2366aade995
6   resourceVersion: '9336'
7   creationTimestamp: '2023-11-02T01:59:01Z'
8   annotations:
9     kubectl.kubernetes.io/last-applied-configuration: >
10    [{"apiVersion":"rbac.authorization.k8s.io/v1","kind":"ClusterRoleBinding
11 > managedFields: ...
24 subjects:
25 - kind: ServiceAccount
26   name: alert-apiserver
27   namespace: acc-global
28 roleRef:
29   apiGroup: rbac.authorization.k8s.io
30   kind: ClusterRole
31   name: alert-apiserver

```

클러스터롤바인딩 삭제

삭제하려는 클러스터롤바인딩을 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 클러스터롤바인딩 이름을 입력하여 삭제한다.

4.2.11.5. 서비스어카운트

서비스어카운트는 쿠버네티스 API 접근 시 파드의 권한을 식별하기 위한 리소스이다.

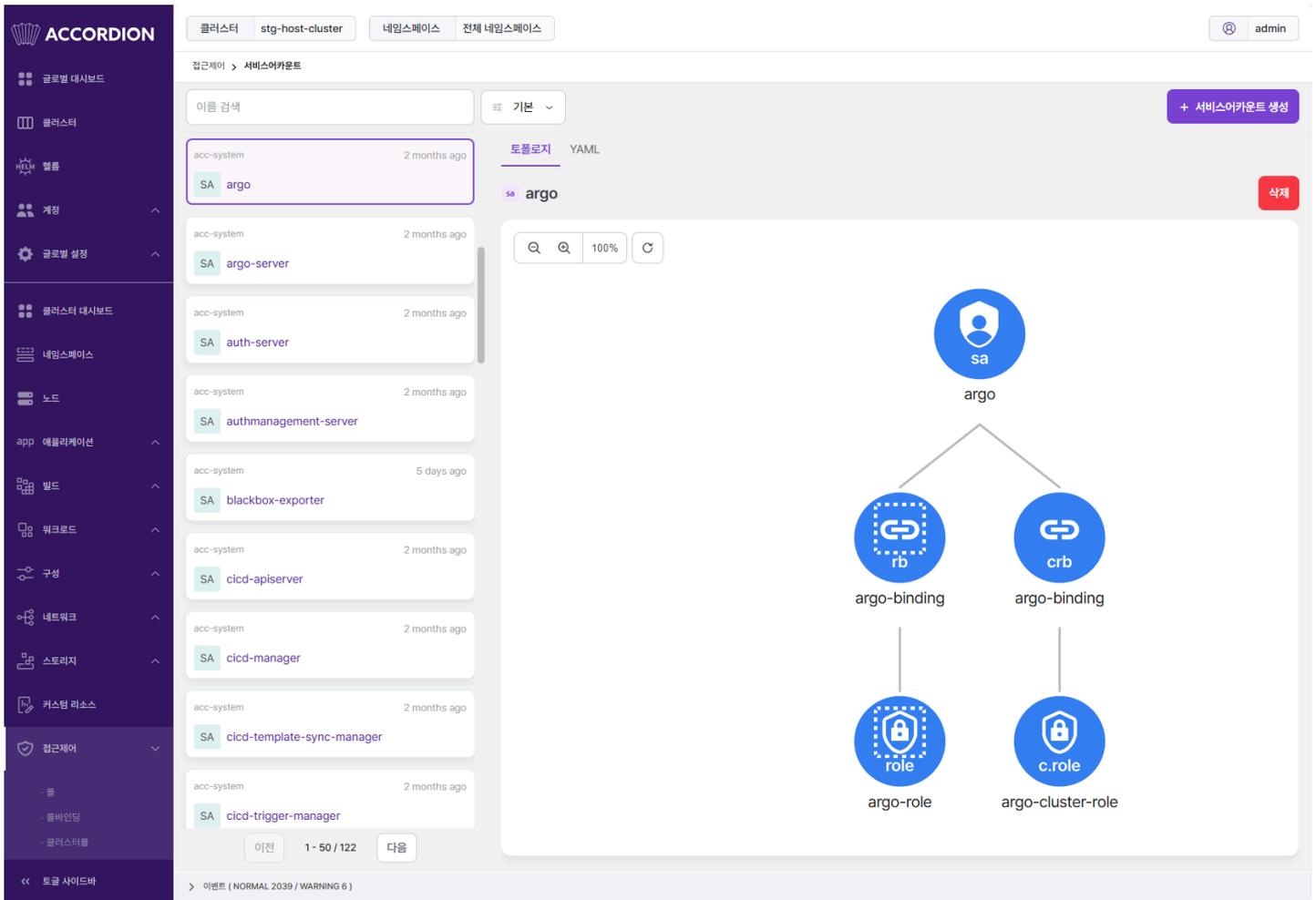
토폴로지

서비스어카운트와 관련된 리소스를 토폴로지 형태로 보여준다.

기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

서비스어카운트의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 서비스어카운트
- 롤바인딩, 클러스터롤바인딩
- 롤, 클러스터롤



서비스어카운트 생성

+ 서비스어카운트 생성 을 선택하면 나타나는 화면에서 쿠버네티스 서비스어카운트 리소스 정보를 입력하여 생성할 수 있다.

접근제어 > 서비스어카운트

← 서비스어카운트 목록

서비스어카운트 생성

```

1  apiVersion: v1
2  kind: ServiceAccount
3  metadata:
4    name: example-serviceaccount
5    namespace: default
6

```

서비스어카운트 수정

수정하려는 서비스어카운트를 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

접근제어 > 서비스어카운트

이름 검색

기본

+ 서비스어카운트 생성

acc-system	1 year ago
SA argo	
acc-system	1 year ago
SA argo-server	
acc-system	1 year ago
SA auth-server	
acc-system	1 year ago
SA authmanagement-server	
acc-system	1 day ago

이전
1 - 50 / 174
다음

토폴로지
YAML

sa argo

수정

삭제

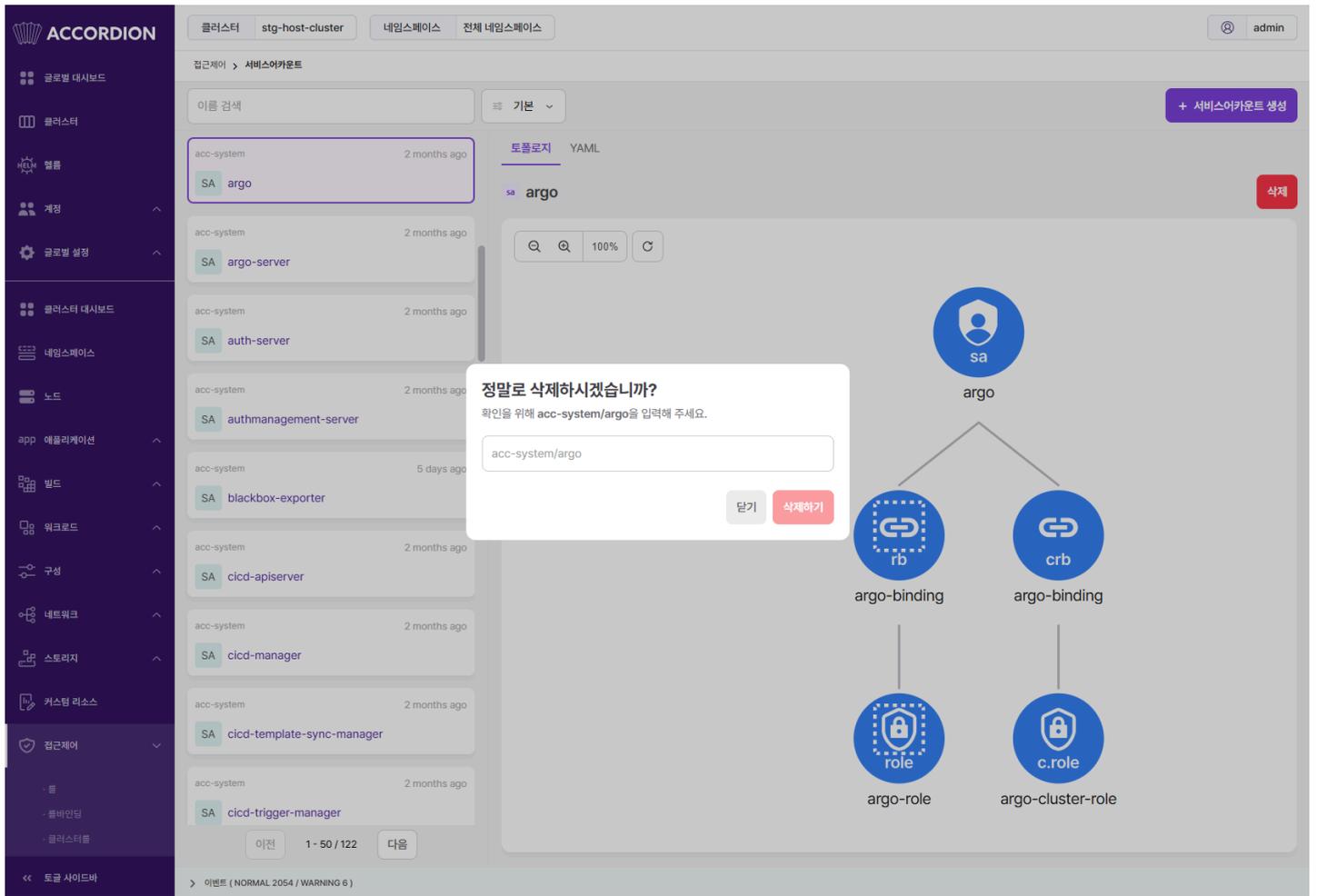
```

1  kind: ServiceAccount
2  apiVersion: v1
3  metadata:
4    name: argo
5    namespace: acc-system
6    uid: a0ac0f0c-5008-4fac-b084-77e858892221
7    resourceVersion: "6368"
8    creationTimestamp: "2023-11-02T01:54:39Z"
9    annotations:
10     kubectl.kubernetes.io/last-applied-configuration: >
11     {"apiVersion":"v1","kind":"ServiceAccount","metadata":{"annotations":{"
12 > managedFields: ...
23

```

서비스어카운트 삭제

삭제하려는 서비스어카운트를 선택하고 우측의 삭제 버튼을 선택한다.



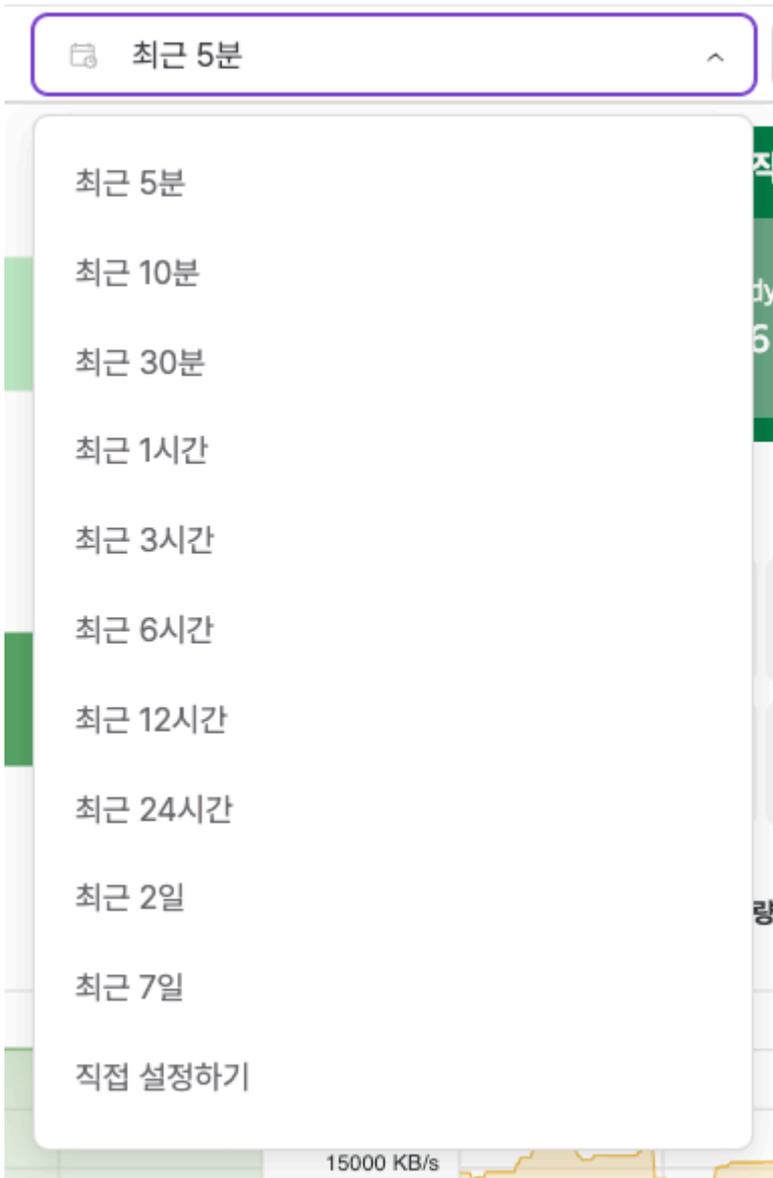
모달에서 네임스페이스와 서비스어카운트 이름을 입력하여 삭제한다.

4.2.12. 모니터링

모니터링은 클러스터 별로 모니터링을 제공한다. 모니터링은 시스템 정보와 이벤트 로그, 감사 로그, 컨테이너 로그 등의 로그 정보 및 서비스 트래픽 정보에 대한 서비스 메시지를 제공하여 운영자가 Accordion 내의 클러스터와 애플리케이션을 다양한 관점에서 모니터링하여 안정적인 시스템 운영할 수 있도록 한다.

모니터링 지표는 시간 별로 검색이 가능하며 아래의 두가지 방법과 같다.

첫째, 최근 시점부터 시간 간격을 설정하여 모니터링 데이터를 확인할 수 있다. 기간은 현재 시점부터 분단위(5,10,30), 시간단위(1,3,6,12,24), 일 단위(2,7)로 지정할 수 있다.



둘째, 직접 설정하기를 선택하여 원하는 검색 날짜/시간의 정보를 확인할 수 있다. 시작 시간 캘린더에서 시작날짜를 선택하고 종료 시간 캘린더에서 종료날짜를 선택한 후 설정을 클릭하여 원하는 검색 기간을 적용한다.

직접 설정하기

시작 시간

2025.01.07 09:52:35

<

2025 ▾

1 ▾

>

일	월	화	수	목	금	토
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

9 ▾

52 ▾

35 ▾

종료 시간

2025.01.07 09:52:35

<

2025 ▾

1 ▾

>

일	월	화	수	목	금	토
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

9 ▾

52 ▾

35 ▾

닫기
설정

4.2.12.1. 시스템

클러스터에서는 기준에 따라 개요, 파드, 노드, 네임스페이스, 디플로이먼트, 스테이트풀셋, 데몬셋, GPU 에 대해 모니터링 지표를 제공한다. 그 중 개요는 클러스터 전체 및 모든 네임스페이스에 대한 리소스 사용률 정보를 제공한다.

모니터링 > 시스템 > 관리자 대시보드

개요
파드
노드
네임스페이스
디플로이먼트
스테이트풀셋
데몬셋
GPU

최근 5분
🔄
🏠

노드당 동작중인 파드의 수

0 — 61

노드당 CPU 사용률

0 — 100

동작중인 노드의 수

Ready
6

Not Ready
0

동작중인 파드의 수

Ready
187

Not Ready
14

디플로이먼트

112
Desired

96
Ready

23
Desired

21
Ready

41
Desired

41
Ready

노드당 동작중인 컨테이너의 수

0 — 65

노드당 메모리 사용률

0 — 100

CPU 사용률

메모리 사용률

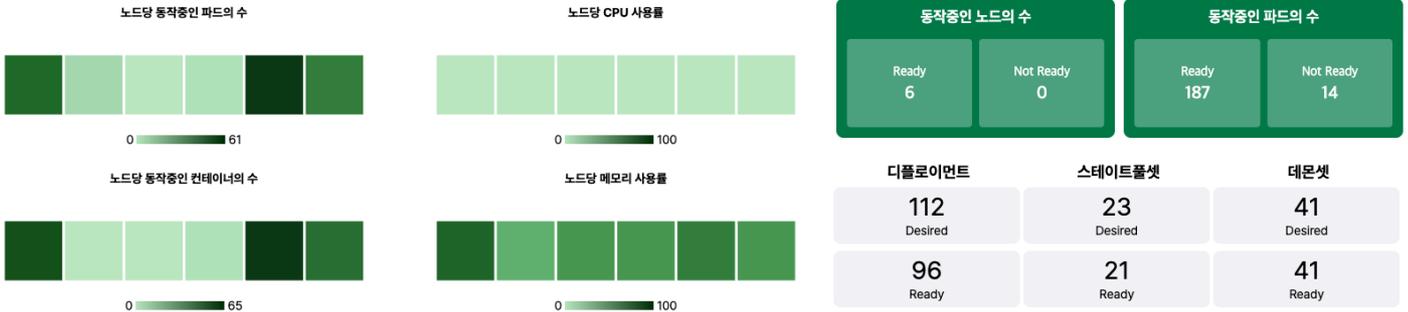
네트워크 사용량

네임스페이스 ↑	CPU 사용량	CPU 요청	CPU 제한	메모리 사용량 (Mi)	메모리 요청 (Mi)	메모리 제한 (Mi)	네트워크 인 (MB/s)	네트워크 아웃 (MB/s)
acc-global	0.28	0	-	3422.83	0	-	6.83	3.07
acc-system	1.46	0	-	26289.56	0	-	25.81	30.86
default	0	0	-	8.11	0	-	0	0
demo	0	0	-	0	0	-	0	0
kafka	0.03	0	-	4177.71	0	-	0	0
kube-node-lease	0	0	-	0	0	-	0	0
kube-public	0	0	-	0	0	-	0	0

호스트맵

클러스터 내 노드와 관련한 정보 및 동작중인 노드, 파드, 디플로이먼트, 스테이트풀셋, 데몬셋에 대한 정보를 보여준다.

자세한 내용은 다음과 같다.



항목	설명
노드당 동작중인 파드의 수	클러스터 내 노드 내 동작중인 파드의 개수를 보여준다. 상대적으로 파드의 개수가 많을 수록 진한 색으로 표시된다.
노드당 CPU 사용률	클러스터 내 노드 내 CPU 사용률을 보여준다. 절대적으로 사용률이 높을수록 진한 색으로 표시된다.
노드당 동작중인 컨테이너의 수	클러스터 내 노드 내 동작중인 컨테이너의 수를 보여준다. 상대적으로 컨테이너의 개수가 많을 수록 진한 색으로 표시된다.
노드당 메모리 사용률	클러스터 내 노드 내 메모리 사용률을 보여준다. 절대적으로 사용률이 높을수록 진한 색으로 표시된다.
동작중인 노드의 수	클러스터 내 노드의 개수를 보여준다. Ready : 동작중인 노드의 개수를 보여준다. Not Ready : 동작중이지 않는 노드의 개수를 보여준다.
동작중인 파드의 수	클러스터 내 파드의 개수를 보여준다. Ready : 동작중인 파드의 개수를 보여준다. Not Ready : 동작중이지 않는 파드의 개수를 보여준다.
디플로이먼트	클러스터 내 디플로이먼트 개수를 보여준다. Desired : 정의된 디플로이먼트 개수를 보여준다. Ready : 동작중인 디플로이먼트 개수를 보여준다.
스테이트풀셋	클러스터 내 스테이트풀셋 개수를 보여준다. Desired : 정의된 스테이트풀셋 개수를 보여준다. Ready : 동작중인 스테이트풀셋 개수를 보여준다.

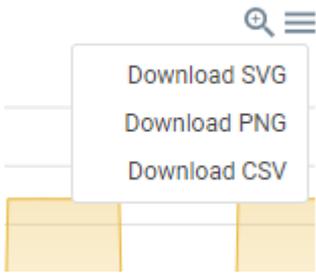
항목	설명
데몬셋	<p>클러스터 내 데몬셋 개수를 보여준다.</p> <p>Desired : 정의된 데몬셋 개수를 보여준다. Ready : 동작중인 데몬셋 개수를 보여준다.</p>

차트

클러스터의 CPU 사용률, 메모리 사용률, 네트워크 사용량과 관련한 정보를 보여준다.



화면 우측 상단의 햄버거 버튼을 선택하면 차트를 다운 받을수 있다.



테이블

클러스터 내 네임스페이스에 대한 정보를 보여준다.

네임스페이스 ↕	CPU 사용량	CPU 요청	CPU 제한	메모리 사용량(MI)	메모리 요청(MI)	메모리 제한(MI)	네트워크 인(MB/S)	네트워크 아웃(MB/S)
acc-global	0.13	0	-	3419.55	0	-	0.06	0.06
acc-system	1.29	0	-	26328.56	0	-	5.04	7.35
default	0	0	-	8.11	0	-	0	0
demo	0	0	-	0	0	-	0	0
kafka	0.03	0	-	4179.48	0	-	0	0
kube-node-lease	0	0	-	0	0	-	0	0
kube-public	0	0	-	0	0	-	0	0
kube-system	1.08	0	-	8100.74	0	-	7.34	12.29
manual	0	0	-	0	0	-	0	0

네임스페이스 정보에 대한 내용은 다음과 같다.

항목	설명
네임스페이스	네임스페이스 이름
CPU 사용량	CPU 사용량
CPU 요청	CPU Quota Request 용량
CPU 제한	CPU Quota Limits 용량
메모리 사용량 (MI)	메모리 사용량
메모리 요청 (MI)	메모리 Quota Request 용량
메모리 제한 (MI)	메모리 Quota Limits 용량
네트워크 인 (MB/S)	Network Inbound 속도

항목	설명
네트워크 아웃 (MB/S)	Network Outbound 속도

파드

클러스터에 배포된 파드의 상태 정보를 제공한다. 노드 또는 네임스페이스별로 필터링하여 파드 정보를 확인할 수 있다.

네임스페이스	파드	상태	노드	호스트 IP	컨테이너	CPU 요청	CPU 제한	메모리 요청	메모리 제한	CPU 사용량	메모리 사용량	수신 데이터	송신 데이터	재발량	AGE
acc-system	acc-kube-state-metrics-6574cc2586-4835	Running	acc-node1	172.32.199.166	3/3	-	-	-	-	4.21 / infinity %	4.100 / infinity %	111 KB/s	37.15 KB/s	0	156
acc-system	acc-node-exporter-6229P	Running	acc-node1	10.20.200.207	2/2	112 m	270 m	200Mi	220Mi	6.11 / infinity %	35.100 / infinity %	291.95 KB/s	584.28 KB/s	2	420
acc-system	acc-node-exporter-9f36	Running	acc-master	10.20.200.206	2/2	112 m	270 m	200Mi	220Mi	6.11 / infinity %	35.100 / infinity %	137.84 KB/s	274.43 KB/s	1	420
acc-system	acc-node-exporter-9p3x	Running	acc-node2	10.20.200.208	2/2	112 m	270 m	200Mi	220Mi	6.11 / infinity %	35.100 / infinity %	166.25 KB/s	208.9 KB/s	2	420
acc-system	accordion-data-provisioner-788b4d88bb-x7555	Running	acc-master	172.32.183.140	1/1	-	-	-	-	2.01 / infinity %	25600 / infinity %	734 Bytes/s	459 Bytes/s	1	420
acc-system	alert-server-0995580f-54c0d	Running	acc-node2	172.32.24.41	1/1	-	-	-	-	0.100 / infinity %	0.100 / infinity %	15 Bytes/s	5 Bytes/s	0	144
acc-system	alertmanager-main-0	Running	acc-node2	172.32.24.33	2/2	-	-	200Mi	-	0.100 / infinity %	0.100 / infinity %	143 Bytes/s	351 Bytes/s	0	156
acc-system	audit-server-5553c475-3a0f5	Running	acc-node1	172.32.199.170	1/1	-	-	-	-	1.07 / infinity %	10.000 / infinity %	28.79 KB/s	14.15 KB/s	0	50
acc-system	blackbox-exporter-84d9648bb-998mm	Running	acc-node1	172.32.199.158	3/3	30 m	60 m	60Mi	120Mi	0.100 / infinity %	0.100 / infinity %	217 Bytes/s	290 Bytes/s	0	156
acc-system	cdp-apiserver-7f47d52f5-Scark	Running	acc-node1	172.32.199.173	1/1	-	-	-	-	1.01 / infinity %	7.888 / infinity %	97 Bytes/s	97 Bytes/s	0	156

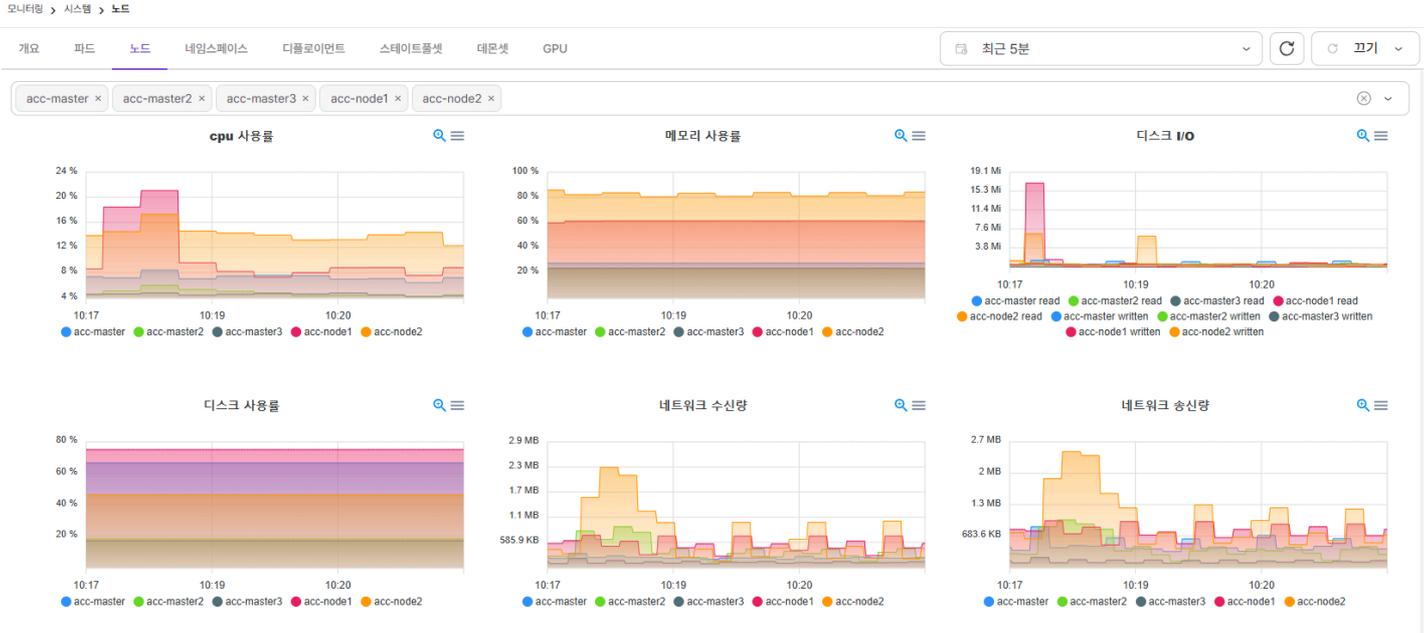
제공하는 정보는 다음과 같다.

항목	설명
네임스페이스	네임스페이스 이름
이름	파드 이름
상태	파드 상태
노드	파드가 배포된 노드 이름
파드 IP	파드 고유 IP
컨테이너	파드의 Running 및 전체 컨테이너 갯수
CPU 요청 / 제한	파드의 CPU Requests/Limits 할당량이다. 파드의 전체 컨테이너가 CPU 리소스 양을 지정한 경우 표시된다. 그외의 경우 "-"로 표시된다.
메모리 요청 / 제한	파드의 Memory 리소스 Requests/Limits 할당량이다. 파드의 전체 컨테이너가 Memory 리소스 양을 지정한 경우 표시된다. 그외의 경우 "-"로 표시된다.
CPU 사용량	파드의 CPU 사용량이며 CPU 리소스 할당량을 제한한 경우 Limits 기준으로 파드의 CPU 사용률을 표시한다. 그외의 경우 파드가 배포된 노드의 CPU 리소스 기준으로 CPU 사용률을 표시한다. CPU 제한이 설정되어있지 않은 경우 infinity가 표시된다.
메모리 사용량	파드의 Memory 사용량이며 Memory 리소스 할당량을 제한한 경우 Limits 기준으로 파드의 Memory 사용률을 표시한다. 그외의 경우 파드가 배포된 노드의 Memory 리소스 기준으로 Memory 사용률을 표시한다. 메모리 제한이 설정되어있지 않은 경우 infinity가 표시된다.
NETWORK RX	파드의 Network Received 속도
NETWORK TX	파드의 Network Transmitted 속도

항목	설명
재실행	파드의 재시작 횟수
AGE	파드의 실행된 시간

노드

노드별 시스템 리소스(CPU, 메모리, 디스크, 네트워크) 사용 정보를 제공한다. 노드의 경우 다중 선택이 가능하며 다중 선택 시 노드별 리소스 사용량 비교가 가능하다.



네임스페이스

네임스페이스별 시스템 리소스(CPU, 메모리) 사용 정보를 제공한다. 노드와 마찬가지로 다중 선택이 가능하며 다중 선택 시 네임스페이스별 리소스 사용량 비교가 가능하다.

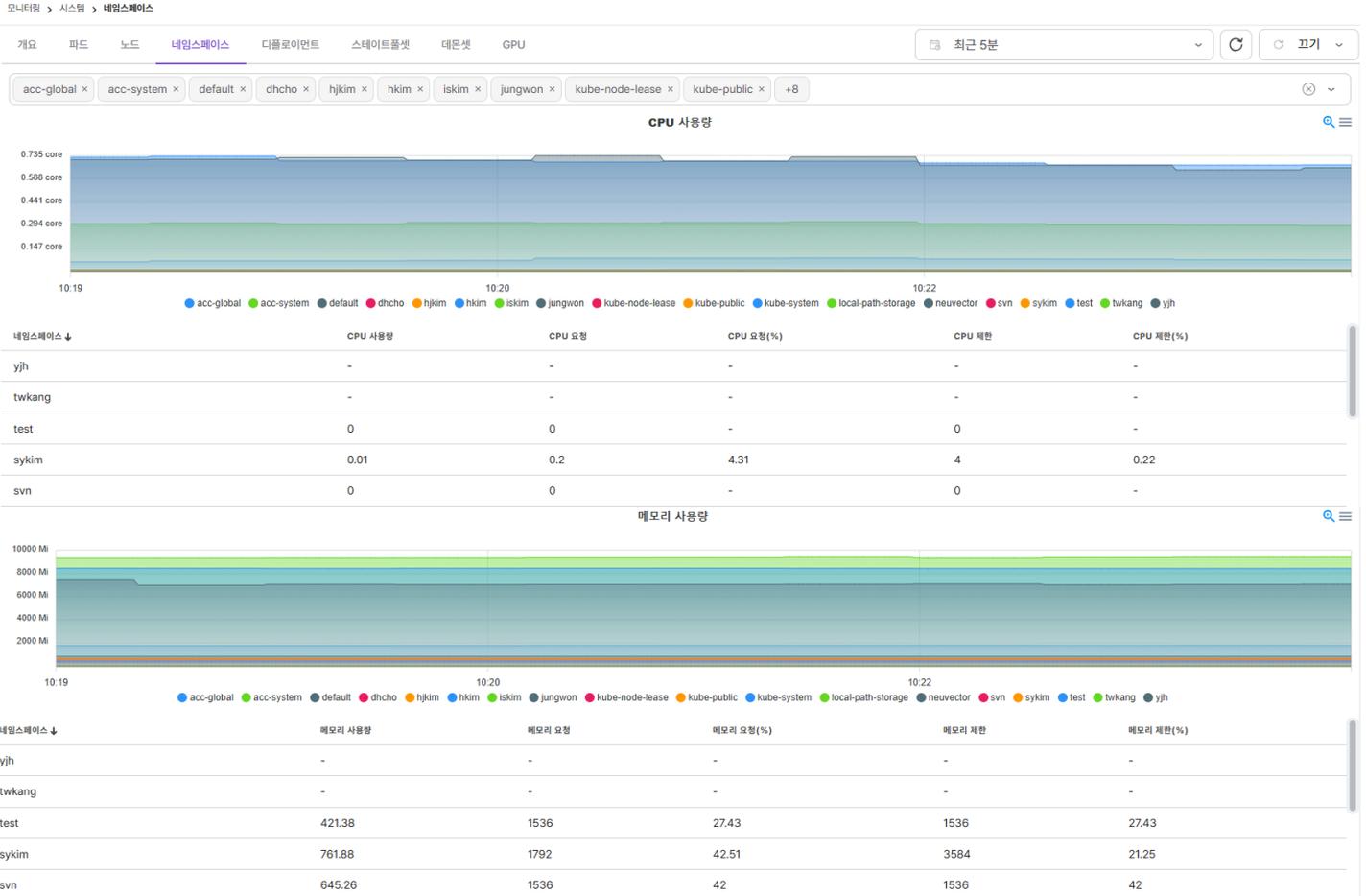


Table 8. CPU

항목	설명
네임스페이스	네임스페이스 이름
CPU 사용량	네임스페이스에 배포된 파드 CPU 사용량 합계
CPU 요청	네임스페이스에 배포된 파드 CPU Requests 합계
CPU 요청(%)	CPU 사용량 / CPU 요청
CPU 제한	네임스페이스에 배포된 파드 CPU Limits 합계
CPU 제한(%)	CPU 사용량 / CPU 제한

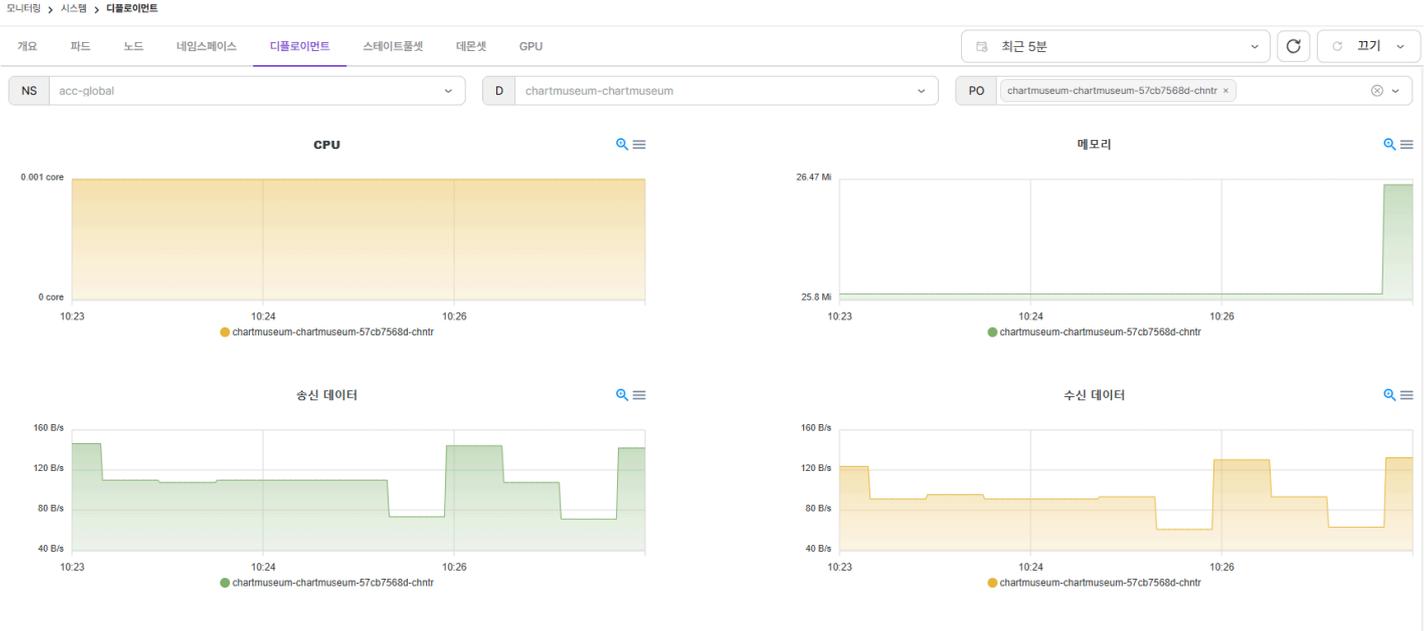
Table 9. 메모리

항목	설명
네임스페이스	네임스페이스 이름
메모리 사용량	네임스페이스에 배포된 파드 메모리 사용량 합계

항목	설명
메모리 요청	네임스페이스에 배포된 파드 메모리 Requests 합계
메모리 요청(%)	메모리 사용량 / 메모리 요청
메모리 제한	네임스페이스에 배포된 파드 메모리 Limits 합계
메모리 제한(%)	메모리 사용량 / 메모리 제한

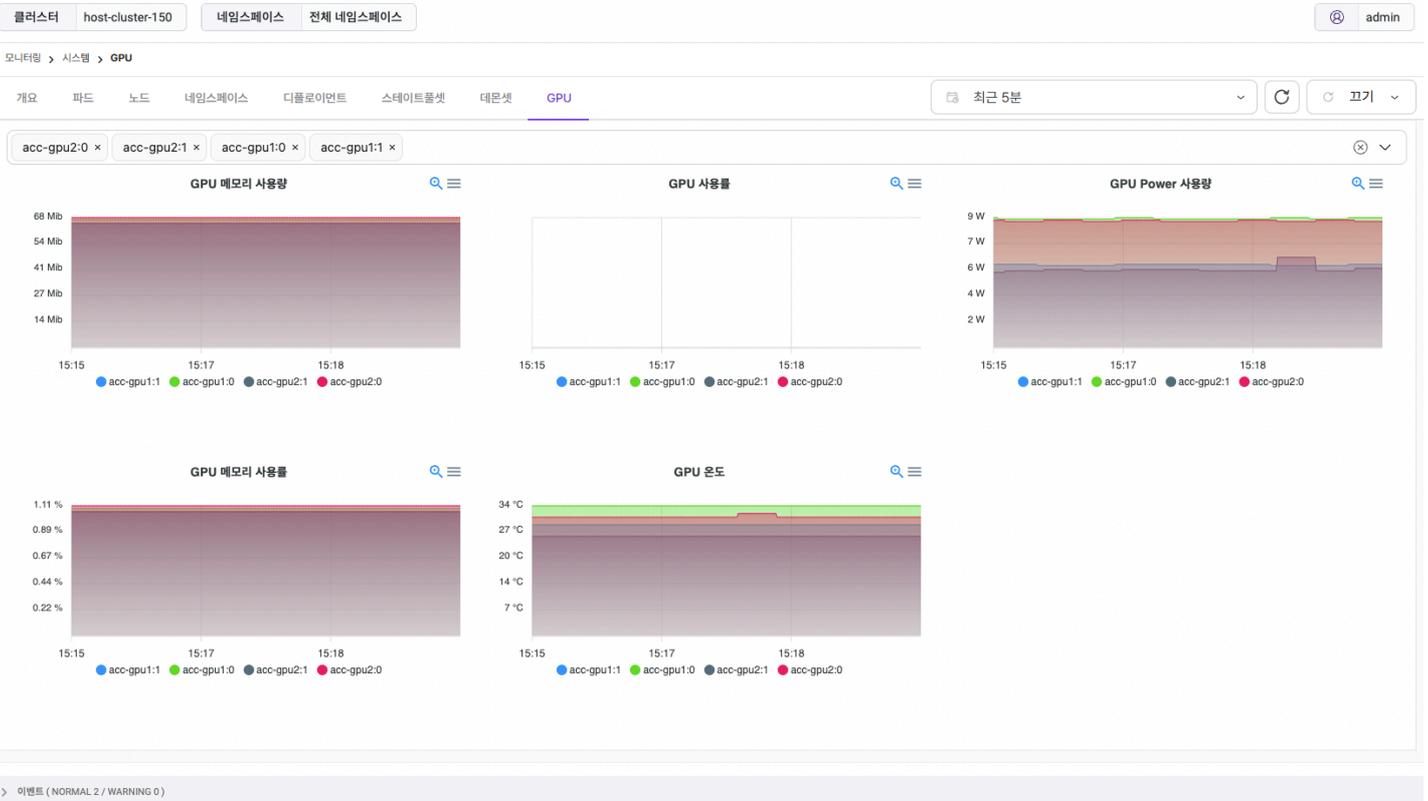
디플로이먼트 / 스테이트풀셋 / 데몬셋

디플로이먼트, 스테이트풀셋, 데몬셋을 기준으로 시스템 리소스 사용 정보를 제공한다. 디플로이먼트, 스테이트풀셋, 데몬셋을 구성하는 개별 파드에 대한 리소스 사용 정보를 확인할 수 있다.



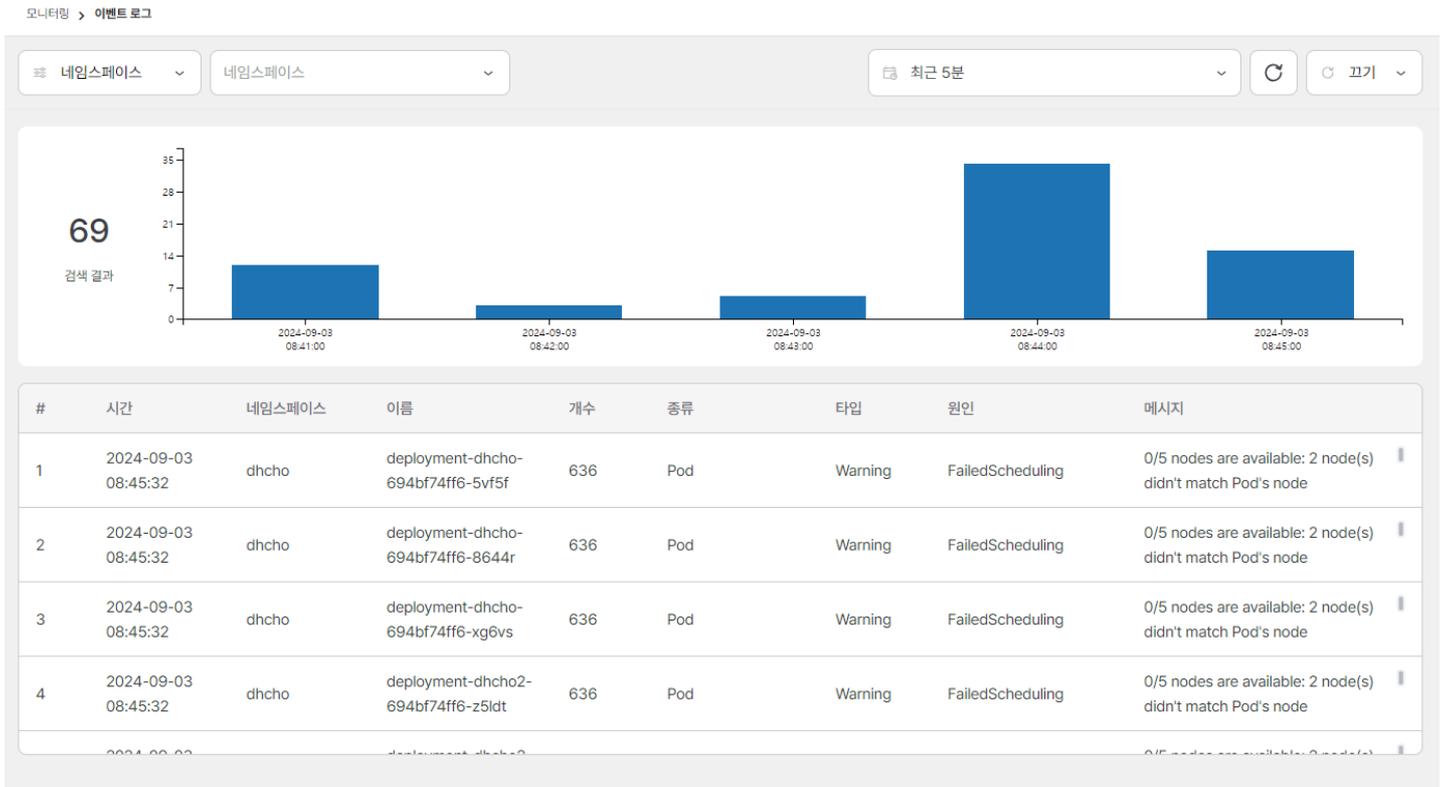
GPU

쿠버네티스 설정상 GPU를 지원하는 경우 GPU 별 리소스 사용 정보를 제공한다.



4.2.12.2. 이벤트 로그

클러스터에서 발생하는 쿠버네티스 이벤트 로그를 제공한다. 시간대별 로그 카운트 정보를 차트로 제공하고 아래 개별 로그에 대한 정보를 제공한다.



제공하는 정보는 다음과 같다.

항목	설명
시간	날짜 (yyyy-mm-dd HH:mm:ss)
네임스페이스	네임스페이스 이름
이름	이벤트 발생 리소스 이름
개수	이벤트 발생 횟수
종류	이벤트 발생 리소스 종류
타입	이벤트 레벨
원인	이벤트 발생 원인
메시지	이벤트에 대한 세부 정보

로그를 선택하면 상세 정보를 확인할 수 있다.

클러스터 host-cluster-200
네임스페이스 전체 네임스페이스

▼ Item {14}
×

모니터링 > 이벤트 로그

네임스페이스
네임스페이스

69

검색 결과

#	시간	네임스페이스	이름	개수	종류
1	2024-09-03 08:45:32	dhcho	deployment-dhcho-694bf74ff6-5vf5f	636	Pod
2	2024-09-03 08:45:32	dhcho	deployment-dhcho-694bf74ff6-8644r	636	Pod
3	2024-09-03 08:45:32	dhcho	deployment-dhcho-694bf74ff6-xg6vs	636	Pod
4	2024-09-03 08:45:32	dhcho	deployment-dhcho2-694bf74ff6-z5ldt	636	Pod

```

kind: Event
apiVersion: v1
metadata: {5}
  name: deployment-dhcho-694bf74ff6-5vf5f.17f0e4c7aecb9679
  namespace: dhcho
  uid: 491bf25d-aed6-459f-967c-5a3ab70b49fa
  resourceVersion: 220737004
  creationTimestamp: 2024-08-31T18:50:03Z
involvedObject: {6}
  kind: Pod
  namespace: dhcho
  name: deployment-dhcho-694bf74ff6-5vf5f
  uid: 2b3fc157-0d7f-41e1-bb4b-22e921b15613
  apiVersion: v1
  resourceVersion: 219186876
reason: FailedScheduling
message: 0/5 nodes are available: 2 node(s) didn't match Pod's node affinity/selector, 3 node(s) had untolerated taint {node-role.kubernetes.io/control-plane: }. preemption: 0/5 nodes are available: 5 Preemption is not helpful for scheduling.
source: {1}
  component: default-scheduler
  firstTimestamp: 2024-08-31T18:50:03Z
  lastTimestamp: 2024-09-02T23:45:32Z
  count: 636
  type: Warning
  eventTime: null
  reportingComponent: default-scheduler
  reportingInstance: value

```

상단의 조건을 설정하여 로그를 필터링할 수 있다. 조건의 항목과 값을 선택하고 돋보기 버튼으로 조건을 설정할 수 있으며 다중 조건 설정이 가능하다. 다중 조건을 설정하면 각각의 조건은 AND 로 조합된다.

클러스터 host-cluster-200
네임스페이스 전체 네임스페이스

최근 5분
🔄
🔍

모니터링 > 이벤트 로그

네임스페이스
네임스페이스

69

검색 결과

#	시간	네임스페이스	이름	개수	종류	타입	원인	메시지
1	2024-09-03 08:45:32	dhcho	deployment-dhcho-694bf74ff6-5vf5f	636	Pod	Warning	FailedScheduling	0/5 nodes are available: 2 node(s) didn't match Pod's node
2	2024-09-03 08:45:32	dhcho	deployment-dhcho-694bf74ff6-8644r	636	Pod	Warning	FailedScheduling	0/5 nodes are available: 2 node(s) didn't match Pod's node
3	2024-09-03 08:45:32	dhcho	deployment-dhcho-694bf74ff6-xg6vs	636	Pod	Warning	FailedScheduling	0/5 nodes are available: 2 node(s) didn't match Pod's node
4	2024-09-03 08:45:32	dhcho	deployment-dhcho2-694bf74ff6-z5ldt	636	Pod	Warning	FailedScheduling	0/5 nodes are available: 2 node(s) didn't match Pod's node

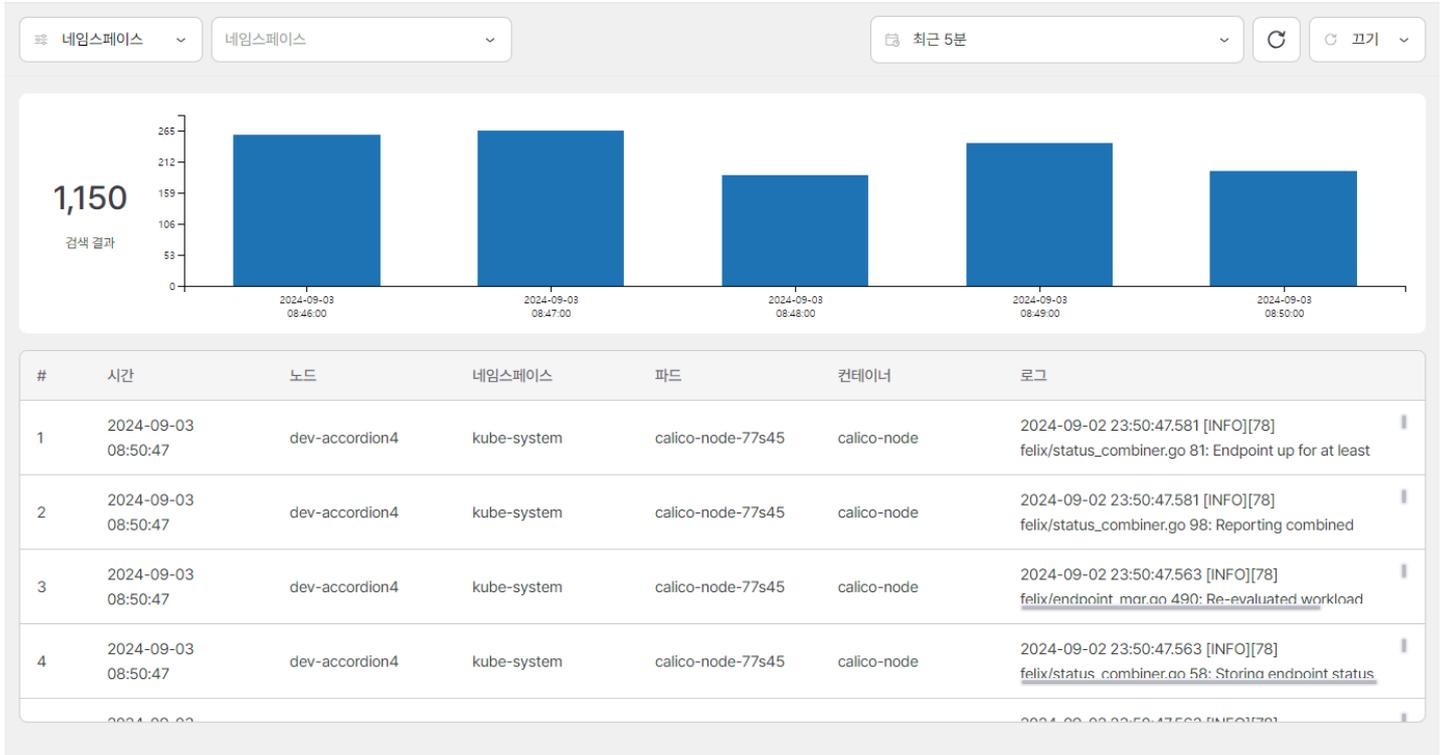
4.2.12.3. 컨테이너 로그

클러스터에 배포된 컨테이너에서 발생한 로그를 제공한다. 시간대별 로그 카운트 정보를 차트로 제공하고 아래 개별 로그에 대한 정보를 제공한다.

TIP

저장 용량의 문제로 모든 네임스페이스에 배포된 컨테이너 로그가 아닌 시스템 네임스페이스인 kube-system 과 acc-system 네임스페이스에 배포된 컨테이너 로그만 제공한다.

모니터링 > 컨테이너 로그



항목	설명
시간	날짜 (yyyy-mm-dd HH:mm:ss)
노드	로그가 발생한 노드
네임스페이스	로그가 발생한 네임스페이스
파드	로그가 발생한 파드
컨테이너	로그가 발생한 컨테이너
로그	로그 메시지

로그를 선택하면 상세 정보를 확인할 수 있다.

클러스터: host-cluster-220 네임스페이스: kube-system 전체 네임스페이스

모니터링 > 컨테이너 로그

네임스페이스: kube-system

#	시간	노드	네임스페이스	파드	컨테이너
1	2024-09-06 08:46:09	acc-master2	kube-system	calico-node-8pdfh	calico-node
2	2024-09-06 08:46:05	acc-master3	kube-system	kube-controller-manager-acc-master3	kube-controller-ma
3	2024-09-06 08:46:05	acc-master3	kube-system	kube-controller-manager-acc-master3	kube-controller-ma
4	2024-09-06 08:46:05	acc-master3	kube-system	kube-controller-manager-acc-master3	kube-controller-ma
5	2024-09-06 08:46:05	acc-master3	kube-system	kube-controller-manager-acc-master3	kube-controller-ma
6	2024-09-06 08:46:05	acc-master3	kube-system	kube-controller-manager-acc-master3	kube-controller-ma
7	2024-09-06 08:46:01	acc-node1	kube-system	calico-node-ztw4j	calico-node
8	2024-09-06 08:45:52	acc-node2	kube-system	calico-node-7qdv	calico-node
9	2024-09-06 08:45:52	acc-node1	kube-system	calico-node-ztw4j	calico-node

> 이벤트 (NORMAL 9 / WARNING 3)

네임스페이스: kube-system 파드: calico-node-8pdfh 컨테이너: calico-node

2024-09-06 08:44:29 to 2024-09-06 08:46:10

```
[0]2024-09-06 08:46:09: 2024-09-05 23:46:09.048 [INFO][94] felix/int_dataplane.go 1689: Received *proto.HostMetadataV4V6Update update from calculation graph #
sg-hostname="acc-master" ipv4_addr:"10.20.200.221/16" labels:<key>"beta.kuberne
tes.io/arch" value:"amd64" > labels:<key>"beta.kubernetes.io/os" value:"linux" >
labels:<key>"kubernetes.io/arch" value:"amd64" > labels:<key>"kubernetes.io/h
ostname" value:"acc-master" > labels:<key>"kubernetes.io/os" value:"linux" > la
bels:<key>"node-role.kubernetes.io/control-plane" value:"" > labels:<key>"node.
kubernetes.io/exclude-from-external-load-balancers" value:"" >
[1]2024-09-06 08:45:52: 2024-09-05 23:45:52.627 [INFO][94] felix/int_dataplane.
go 1689: Received *proto.HostMetadataV4V6Update update from calculation graph #
sg-hostname="acc-node1" ipv4_addr:"10.20.200.224/16" labels:<key>"beta.kubernet
es.io/arch" value:"amd64" > labels:<key>"beta.kubernetes.io/os" value:"linux" >
labels:<key>"kubernetes.io/arch" value:"amd64" > labels:<key>"kubernetes.io/hc
stname" value:"acc-node1" > labels:<key>"kubernetes.io/os" value:"linux" > labe
ls:<key>"node-role.kubernetes.io/infra" value:"" > labels:<key>"node-role.kuber
netes.io/worker" value:"" >
[2]2024-09-06 08:45:41: 2024-09-05 23:45:41.782 [INFO][92] monitor-addresses/au
todetection_methods.go 117: Using autodetected IPv4 address 10.20.200.222/16 on
matching interface eni192
[3]2024-09-06 08:45:41: 2024-09-05 23:45:41.536 [INFO][94] felix/int_dataplane.
go 1689: Received *proto.HostMetadataV4V6Update update from calculation graph #
sg-hostname="acc-node2" ipv4_addr:"10.20.200.225/16" labels:<key>"beta.kubernet
es.io/arch" value:"amd64" > labels:<key>"beta.kubernetes.io/os" value:"linux" >
labels:<key>"kubernetes.io/arch" value:"amd64" > labels:<key>"kubernetes.io/hc
stname" value:"acc-node2" > labels:<key>"kubernetes.io/os" value:"linux" > labe
ls:<key>"node-role.kubernetes.io/infra" value:"" > labels:<key>"node-role.kubet
netes.io/worker" value:"" >
[4]2024-09-06 08:45:41: 2024-09-05 23:45:41.536 [INFO][94] felix/summary.go 100
: Summarising 11 dataplane reconciliation loops over 1m2.6s: avg=4ms longest=12
ms ()
[5]2024-09-06 08:44:41: 2024-09-05 23:44:41.781 [INFO][92] monitor-addresses/au
todetection_methods.go 117: Using autodetected IPv4 address 10.20.200.222/16 on
matching interface eni192
[6]2024-09-06 08:44:38: 2024-09-05 23:44:38.894 [INFO][94] felix/summary.go 100
: Summarising 8 dataplane reconciliation loops over 1m1.3s: avg=7ms longest=36m
s (resync-net-v4)
```

상단의 조건을 설정하여 로그를 필터링할 수 있다. 조건의 항목과 값을 선택하고 돋보기 버튼으로 조건을 설정할 수 있으며 다중 조건 설정이 가능하다. 다중 조건을 설정하면 각각의 조건은 AND 로 조합된다.

모니터링 > 컨테이너 로그

🔍 네임스페이스

네임스페이스

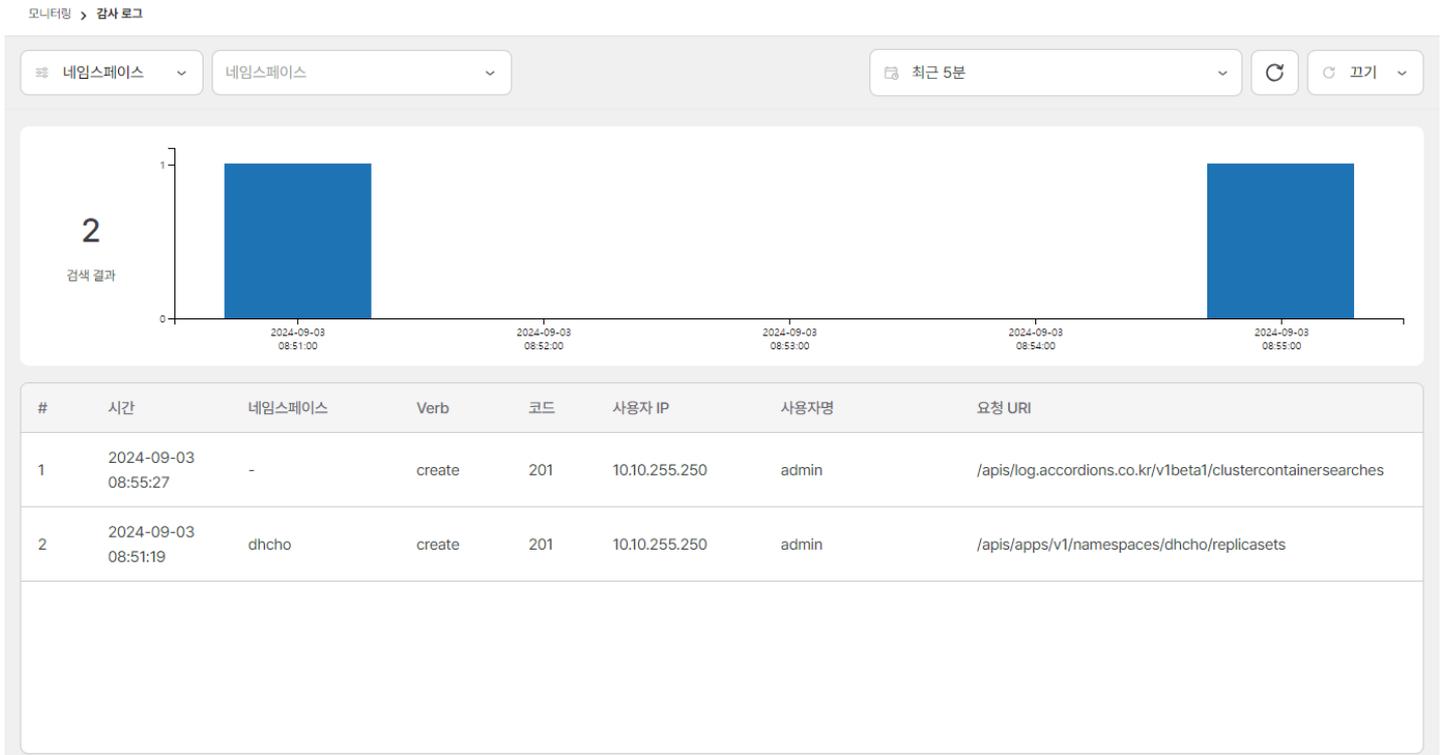
🕒 최근 5분
🔄
🗑️ 끄기

네임스페이스
 노드명
 파드명
 컨테이너명
 로그

#	시간	노드	네임스페이스	파드	컨테이너	로그
1	2024-09-06 08:46:09	acc-master2	kube-system	calico-node-8pdfh	calico-node	2024-09-05 23:46:09.048 [INFO][94] felix/int_dataplane.go 1689: Received *proto.HostMetadata/v4V6Update update from calculation
2	2024-09-06 08:46:05	acc-master3	kube-system	kube-controller-manager-acc-master3	kube-controller-manager	10906 08:46:05.061107 1 event.go:376] "Event occurred" object="haproxy/haproxy-test-7d747864-25db4859" fieldPath=""
3	2024-09-06 08:46:05	acc-master3	kube-system	kube-controller-manager-acc-master3	kube-controller-manager	10906 08:46:05.061362 1 event.go:376] "Event occurred" object="aaaa/www-web-0" kind="PersistentVolumeClaim" fieldPath=""
4	2024-09-06 08:46:05	acc-master3	kube-system	kube-controller-manager-acc-master3	kube-controller-manager	10906 08:46:05.061513 1 event.go:376] "Event occurred" object="sykim/egw-target-8ac8c2f2-81b07cff" fieldPath=""
5	2024-09-06 08:46:05	acc-master3	kube-system	kube-controller-manager-acc-master3	kube-controller-manager	10906 08:46:05.061740 1 event.go:376] "Event occurred" object="jenkins/test-jenkins-82ca3b73-0970cd68" fieldPath=""
6	2024-09-06 08:46:05	acc-master3	kube-system	kube-controller-manager-acc-master3	kube-controller-manager	10906 08:46:05.061768 1 event.go:376] "Event occurred" object="thlee/data-jupyterhub-postgresql-0" fieldPath=""
7	2024-09-06 08:46:01	acc-node1	kube-system	calico-node-ztw4j	calico-node	2024-09-05 23:46:01.923 [INFO][63] felix/summary.go 100: Summarising 14 dataplane reconciliation loops over 1m8.7s: avg=13ms
8	2024-09-06 08:45:52	acc-node2	kube-system	calico-node-7qdv	calico-node	2024-09-05 23:45:52.628 [INFO][74] felix/int_dataplane.go 1689: Received *proto.HostMetadata/v4V6Update update from calculation
9	2024-09-06 08:45:52	acc-node1	kube-system	calico-node-ztw4j	calico-node	2024-09-05 23:45:52.627 [INFO][63] felix/int_dataplane.go 1689: Received *proto.HostMetadata/v4V6Update update from calculation

4.2.12.4. 감사 로그

클러스터에 배포된 쿠버네티스 리소스에 대한 접근 감사 로그를 제공한다. 시간대별 카운트 정보를 차트로 제공하고 아래 개별 로그에 대한 정보를 제공한다.



항목	설명
시간	날짜 (yyyy-mm-dd HH:mm:ss)
네임스페이스	사용자가 접근한 리소스가 배포된 네임스페이스
VERB	리소스 접근 요청시 사용한 API 동사 (get, list, create, update, patch, watch, delete)
코드	리소스 접근 요청에 대한 응답 HTTP 코드
사용자 IP	요청이 발생한 IP
사용자명	리소스 접근 요청을 한 사용자
요청 URI	리소스 접근 요청 엔드포인트

로그를 선택하면 상세 정보를 확인할 수 있다.

클러스터 host-cluster-200 네임스페이스 전체 네임스페이스

모니터링 > 감사 로그

네임스페이스 네임스페이스

3
검색 결과

#	시간	네임스페이스	Verb	코드	사용자 IP	사용
1	2024-09-03 08:56:27	-	create	201	10.10.255.250	adr
2	2024-09-03 08:56:22	-	create	201	10.10.255.250	adr
3	2024-09-03 08:55:27	-	create	201	10.10.255.250	adr

```

Item {17}
  kind: Audit
  apiVersion: log.accordions.co.kr/v1beta1
  metadata {2}
    uid: c8c37829-3f79-413e-bf7f-31867a13cde8
    creationTimestamp: 2024-09-02T23:56:27Z
  level: Metadata
  auditID: c8c37829-3f79-413e-bf7f-31867a13cde8
  stage: ResponseComplete
  requestURI: /apis/log.accordions.co.kr/v1beta1/clustercontainersearches
  verb: create
  user {4}
    username: system:serviceaccount:acc-system:member-agent
    uid: cf9b1b1f-6965-426b-859e-d4a9db4bfd9
    groups {3}
      0: system:serviceaccounts
      1: system:serviceaccounts:acc-system
      2: system:authenticated
    extra {2}
      authentication.kubernetes.io/pod-name [1]
        0: member-agent-5f67b7c848-b417p
      authentication.kubernetes.io/pod-uid [1]
        0: b96107d6-6adc-4733-88db-8b9265cb9822
    impersonatedUser {2}
      username: accordion:user:admin
      groups {1}
        0: system:authenticated
    sourceIPs {5}
      0: 10.10.255.250
      1: 172.32.190.62
      2: 10.20.200.202
      3: 10.20.200.201
      4: 10.20.200.205
  
```

상단의 조건을 설정하여 로그를 필터링할 수 있다. 조건의 항목과 값을 선택하고 돋보기 버튼으로 조건을 설정할 수 있으며 다중 조건 설정이 가능하다. 다중 조건을 설정하면 각각의 조건은 AND 로 조합된다.

모니터링 > 감사 로그

네임스페이스 네임스페이스 최근 5분 🔍 🔄 🗑️

네임스페이스

사용자명

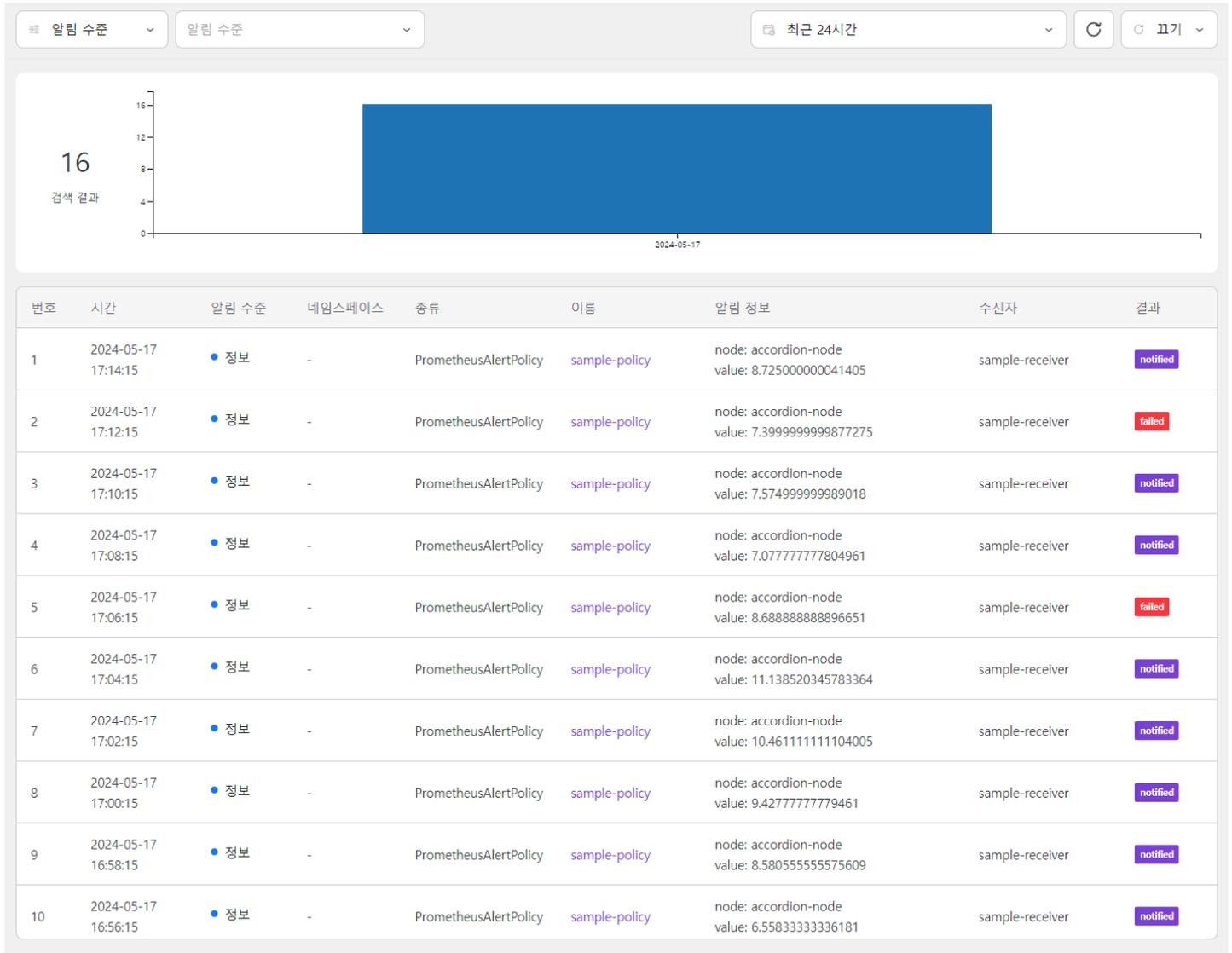
Verb

코드

#	시간	네임스페이스	Verb	코드	사용자 IP	사용자명	요청 URI
1	2024-09-03 08:55:27	-	create	201	10.10.255.250	admin	/apis/log.accordions.co.kr/v1beta1/clustercontainersearches
2	2024-09-03 08:51:19	dhcho	create	201	10.10.255.250	admin	/apis/apps/v1/namespaces/dhcho/replicasets

4.2.12.5. 알림 로그

클러스터에서 발생하는 쿠버네티스 알림 로그를 제공한다. 시간대별 로그 카운트 정보를 차트로 제공하고 하단 테이블 형태로 개별 로그에 대한 정보를 제공한다.



제공하는 정보는 다음과 같다.

항목	설명
시간	날짜 (yyyy-mm-dd HH:mm:ss)
알림 수준	알림 경보 수준
네임스페이스	네임스페이스 이름
종류	알림 발생 리소스 종류
이름	알림 발생 리소스 이름
알림 정보	알림 발생 정보
수신자	알림을 받는 수신자 목록

항목	설명
결과	알림 발송 결과

로그를 선택하면 상세 정보를 확인할 수 있다.

클러스터 host-cluster-200 네임스페이스 전체 네임스페이스

모니터링 > 알림 로그

알림 수준 [정보] 알림 수준 [정보]

번호	시간	알림 수준	네임스페이스	종류	이름
1	2024-05-17 17:14:15	정보	-	PrometheusAlertPolicy	sample-policy
2	2024-05-17 17:12:15	정보	-	PrometheusAlertPolicy	sample-policy
3	2024-05-17 17:10:15	정보	-	PrometheusAlertPolicy	sample-policy
4	2024-05-17 17:08:15	정보	-	PrometheusAlertPolicy	sample-policy
5	2024-05-17 17:06:15	정보	-	PrometheusAlertPolicy	sample-policy
6	2024-05-17 17:04:15	정보	-	PrometheusAlertPolicy	sample-policy
7	2024-05-17 17:02:15	정보	-	PrometheusAlertPolicy	sample-policy
8	2024-05-17 17:00:15	정보	-	PrometheusAlertPolicy	sample-policy
9	2024-05-17 16:58:15	정보	-	PrometheusAlertPolicy	sample-policy
10	2024-05-17 16:56:15	정보	-	PrometheusAlertPolicy	sample-policy

> 이벤트 (NORMAL 32 / WARNING 30)

```

Item {17}
kind : Audit
apiVersion : log.accordions.co.kr/v1beta1
metadata {2}
  uid : beb11faa-c2ec-4748-a9e3-e135445e22c8
  creationTimestamp : 2024-05-17T08:14:15Z
level : Request
auditID : beb11faa-c2ec-4748-a9e3-e135445e22c8
stage : ResponseComplete
requestU : /apis/alert.accordions.co.kr/v1beta1/Notifications/sample-policy-450220339
RI
verb : update
user {4}
  usernam : system:serviceaccount:acc-system:alertmanagement-server
  uid : c779fff4-9982-4bea-8c00-96d8dead2e08
  groups {3}
    0 : system:serviceaccounts
    1 : system:serviceaccounts:acc-system
    2 : system:authenticated
  extra {2}
    authentication.kubernetes.io/pod-name {1}
      0 : alertmanagement-server-5b5bf689bb-m28v5
    authentication.kubernetes.io/pod-uid {1}
      0 : 98fc7068-a74c-4b8d-baac-eea0cb35108c
  sourceIPs {1}
    0 : 10.20.200.224
  userAgent : alertmanagement-server/v0.0.0 (linux/amd64)
  t
  objectRef {6}
    resource : Notifications
    name : sample-policy-450220339
    uid : 2714f9c6-4c09-4609-9d6b-f79db384e091
    apiGroup : alert.accordions.co.kr
    apiVersion : v1beta1
    resourceVersion : 47157777
  responseStatus {2}
    metadata {0}
      (empty object)
    code : 200
  requestObject {5}
    apiVersion : alert.accordions.co.kr/v1beta1
    kind : Notification
    metadata {6}
      creationTimestamp : 2024-05-17T08:14:14Z
      generation : 1
      labels {1}
        created_by : alertmanagement-server
      name : sample-policy-450220339
      resourceVersion : 47157777
      uid : 2714f9c6-4c09-4609-9d6b-f79db384e091
    spec {5}
      alerts {1}
  
```

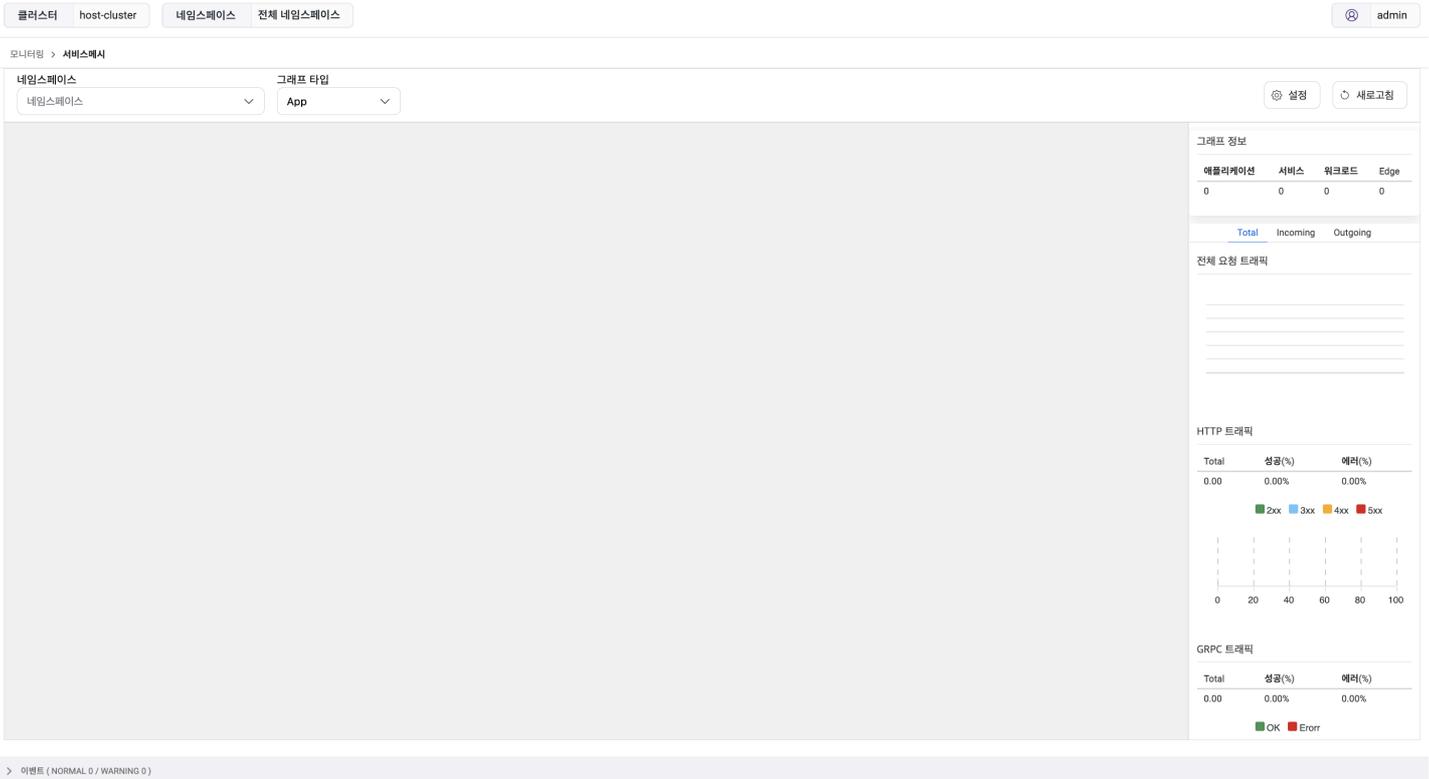
상단의 좌측의 조건을 설정하여 로그를 필터링할 수 있다. 조건의 항목과 값을 선택 및 검색 버튼으로 추가할 수 있으며 다중 조건 설정이 가능하다. 다중 조건을 설정하면 각각의 조건은 AND 로 조합된다.

결과 [정보]

알림 수준: 위험 × 종류: ClusterPromethe... × 이름: sample × 알림 정보: node × 수신자: sample-receiver ×

4.2.12.6. 서비스메시

서비스메시는 서비스 간의 통신을 제어하고 표시하고 관리하는 기능을 의미한다. 현재 아코디언에서는 서비스메시 메뉴에서 서비스메시의 다양한 기능 중 서비스 간의 통신을 표시하는 기능을 제공하고 있다.



상단의 네임스페이스 항목에 모니터링을 원하는 네임스페이스 목록을 선택하면 해당 네임스페이스 내에서 발생하는 서비스 간의 통신 상태를 그래프로 확인할 수 있고 그래프 타입을 변경하여 관점의 기준을 애플리케이션이나 애플리케이션 버전, 서비스, 워크로드 등으로 변경할 수 있다.

TIP

네임스페이스는 다중 선택이 가능하다.

클러스터 host-cluster 네임스페이스 전체 네임스페이스 admin

모니터링 > 서비스메시

네임스페이스 test 그래프 타입 App

설정 새로고침

그래프 정보

애플리케이션	서비스	워크로드	Edge
3	1	0	4

Total Incoming Outgoing

전체 요청 트래픽

HTTP 트래픽

Total	성공(%)	에러(%)
0.49	100.00%	0.00%

2xx 3xx 4xx 5xx

0 20 40 60 80 100

GRPC 트래픽

Total	성공(%)	에러(%)
0.00	0.00%	0.00%

OK Error

> 이벤트 (NORMAL 0 / WARNING 0)

우측 상단의 설정 버튼을 선택하면 트래픽 정보, 갱신 시간, 디스플레이 옵션 등을 설정할 수 있다.

클러스터 host-cluster 네임스페이스 전체 네임스페이스 admin

모니터링 > 서비스메시

네임스페이스 test 그래프 타입 App

설정 새로고침

그래프 정보

애플리케이션	서비스	워크로드	Edge
3	1	0	4

Total Incoming Outgoing

전체 요청 트래픽

HTTP 트래픽

Total	성공(%)	에러(%)
0.51	100.00%	0.00%

2xx 3xx 4xx 5xx

0 20 40 60 80 100

GRPC 트래픽

Total	성공(%)	에러(%)
0.00	0.00%	0.00%

OK Error

설정

트래픽 보기: Requests per Seconds

최근 요청: 최근 5분

인터벌 시간: 매 10초

디스플레이 옵션: Namespace x

닫기

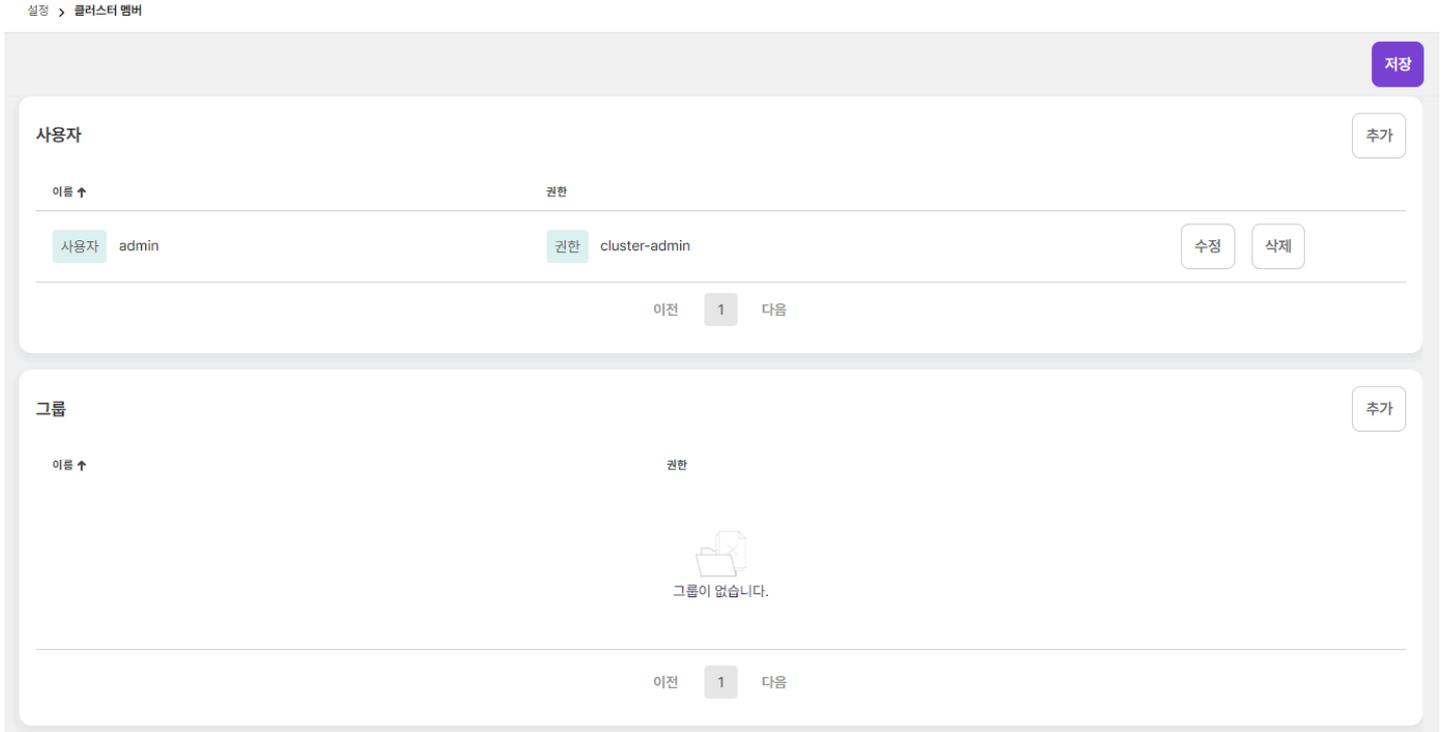
> 이벤트 (NORMAL 0 / WARNING 0)

4.2.13. 설정

설정에는 클러스터 동작에 필요한 정보를 관리하며 클러스터 공통으로 적용된다. 이미지 저장을 위한 레지스트리 설정 및 알림 설정, 클러스터 권한을 설정할 수 있다.

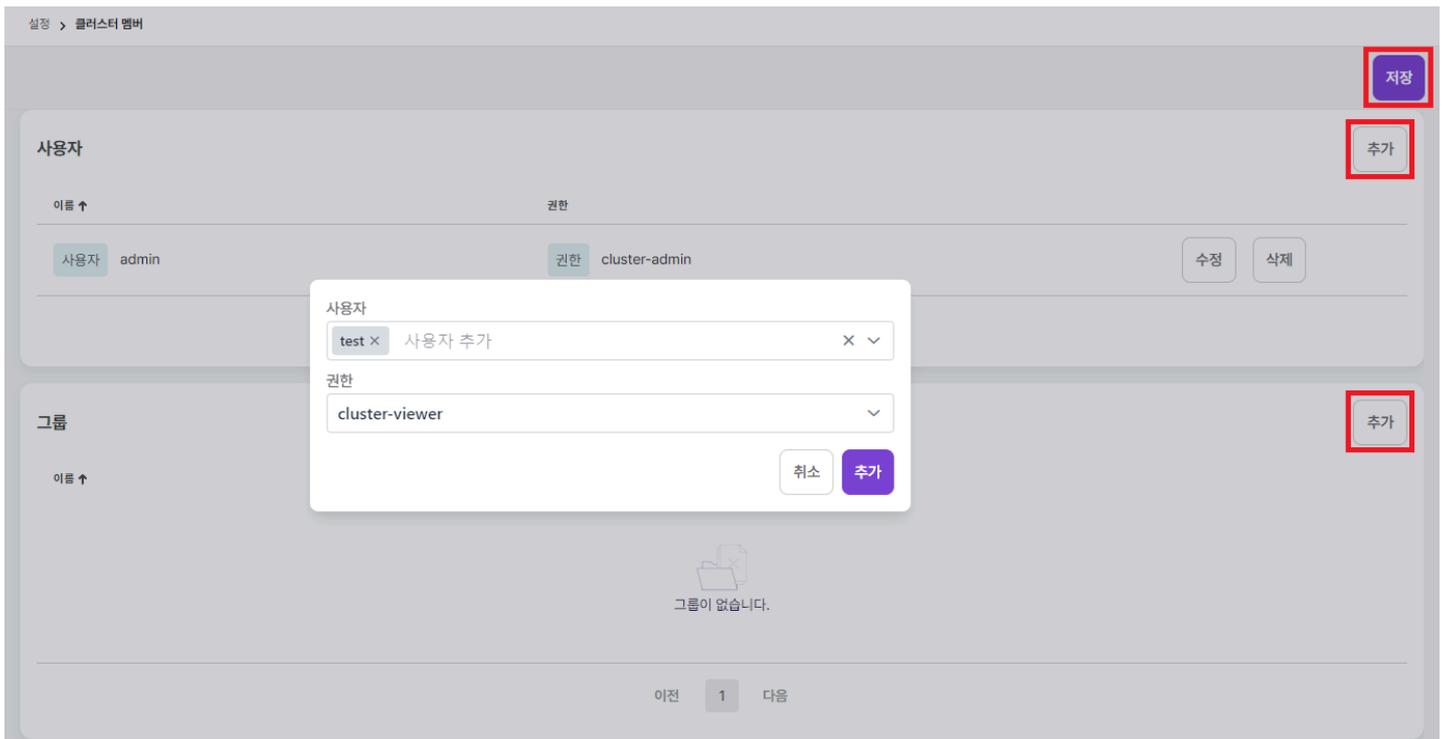
4.2.13.1. 클러스터 멤버

클러스터 사용자와 그룹의 권한을 관리한다.



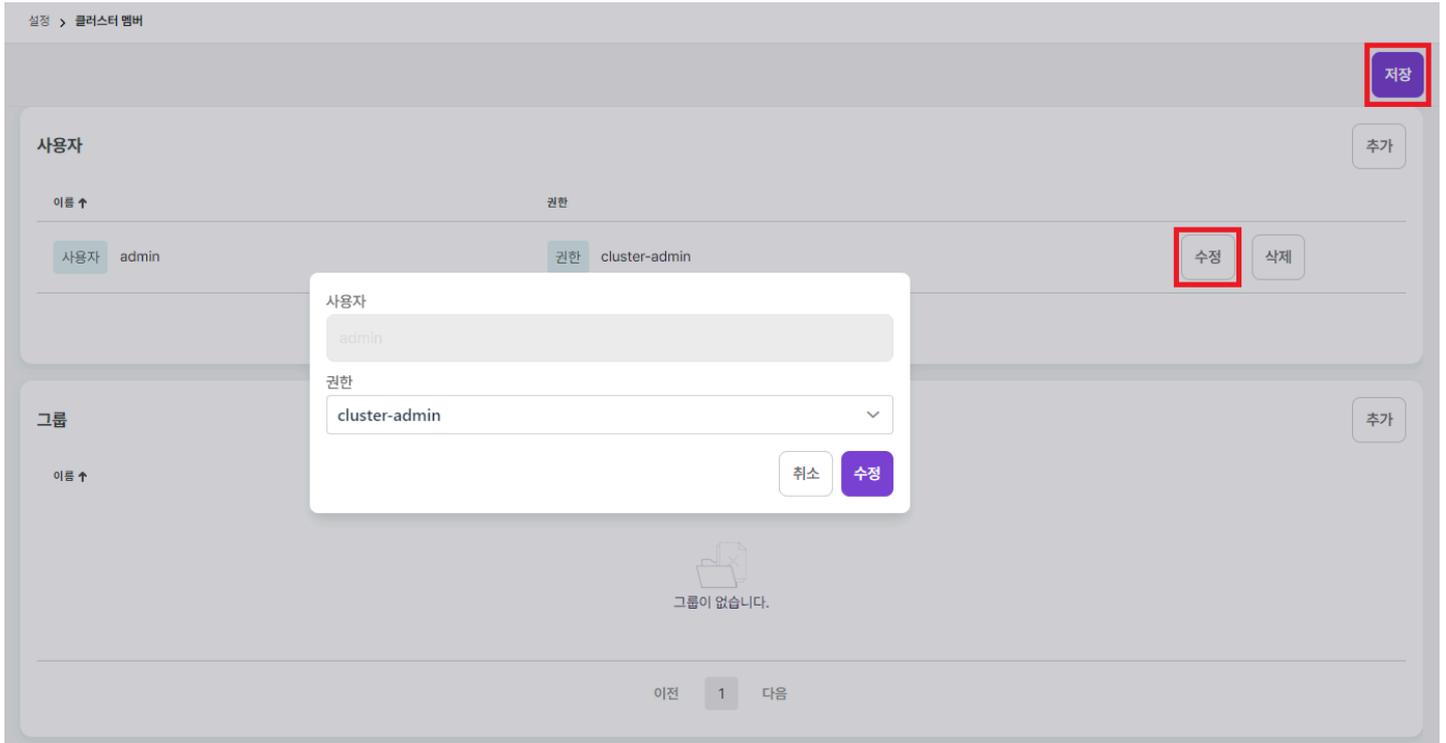
클러스터 멤버 추가

사용자(또는 그룹) 목록 우측의 추가 버튼을 선택한다. 모달에서 사용자(또는 그룹)을 선택해 권한을 설정할 수 있다. 설정 후에는 저장 버튼을 선택해야 변경 사항이 반영된다. 사용자(또는 그룹)은 다중 선택이 가능하지만 권한은 단일 선택만이 가능하다.



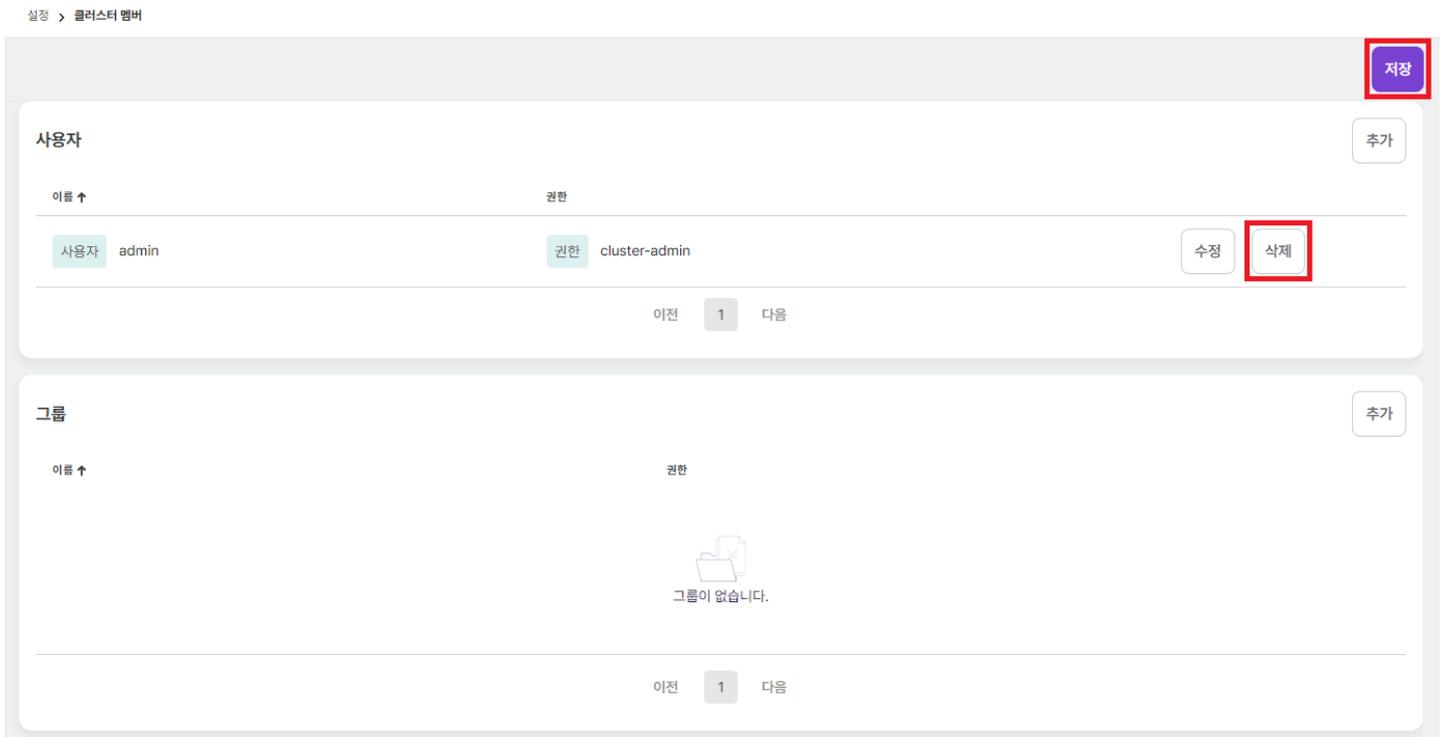
클러스터 멤버 수정

사용자(또는 그룹) 우측의 수정 버튼으로 해당 사용자(또는 그룹)의 권한을 수정할 수 있다. 이후 상단의 저장 버튼을 클릭해야만 변경 사항이 반영된다.



클러스터 멤버 삭제

사용자(또는 그룹)의 우측 삭제 버튼을 선택하면 멤버를 삭제할 수 있다. 이후 상단의 저장 버튼을 클릭해야만 변경 사항이 반영된다.



4.2.13.2. 클러스터 수신자

클러스터 스코프의 클러스터 수신자 메뉴는 글로벌 스코프의 글로벌 수신자 메뉴와 사용 방법이 동일하기 때문에 해당 가이드를 참고한다.

4.2.13.3. 알림 정책

알림 정책은 시스템 성능에 대한 알림의 기준을 설정한다. 사용자가 정한 알림의 기준을 벗어나는 상황이 발생하게 되는 경우 사용자가 빠르게 대처할 수 있도록 알림을 발송한다. 클러스터 스코프의 알림 정책에서는 노드와 워크로드에 관련된 알림을 설정할 수 있고 네임스페이스 스코프의 알림 정책에서는 워크로드에 관련된 알림을 설정할 수 있다.

TIP

알림 정책의 기준은 사용자가 임의로 정의하여 설정할 수 있다.

● Inactive

sample-alertpolicy

Sample Setting Alertpolicy

🕒 2024-05-23 11:18:27

수정
삭제

sample-alertpolicy

알림 활성화: ● Inactive

알림 이름: **sample-alertpolicy**

알림 설명: **Sample Setting Alertpolicy**

알림 규칙

알림 규칙 이름: **sample-policy1**

알림 규칙 설명: **rule1**

알림 대상: **대상 지정**

 메트릭 타입: **CPU 사용률**

 오브젝트

 종류: **노드**

 이름: **dev-accordion1**

임계 조건

 연산자: **일치**

 임계치: **100%**

알림 수준: **정보**

알림 주기

 그룹 대기 시간: **5m**

 그룹 반복 시간: **5m**

 알림 반복 시간: **5m**

일시 정지

 스케줄: **0 0 1 * ***

 지속시간: **5h**

알림 규칙 이름: **sample-policy2**

알림 규칙 설명: **rule2**

알림 대상: **대상 지정**

 메트릭 타입: **메모리 사용률**

 오브젝트

 종류: **디플로이먼트**

 네임스페이스: **sample**

 라벨 선택자: **app = sample**

임계 조건

 연산자: **일치**

 임계치: **100%**

알림 수준: **경고**

알림 수준: **정보**

알림 주기

 그룹 대기 시간: **5s**

 그룹 반복 시간: **5m**

 알림 반복 시간: **5h**

일시 정지

 스케줄: **0 0 * * 0**

 지속시간: **5h**

수신자 목록:

- **sample-receiver**

알림 정책 생성

+ 알림 생성 버튼을 선택하면 알림을 설정할 수 있는 폼이 나타난다. 폼에서 설정 정보를 입력하고 알림 생성 버튼을 선택하면 알림 정책이 생성된다. 하나의 알림 정책은 다수의 알림 규칙들을 가질 수 있으며 각 규칙은 다른 알림 대상, 알림 수준, 알림 주기, 일시 정지 등을 설정할 수 있다.

알림 정책

상단부와 하단부에 위치한 알림 정책 폼에 대한 설정 내용을 설명한다.

← 알림 목록

알림 생성

알림 활성화

이름 설명

항목	설명
활성화	알림 정책 활성화 여부
알림 정책명	알림 정책 이름
알림 정책 설명	알림 정책에 대한 설명

알림 수준
 위험 경고 정보

알림 주기

그룹 대기 시간: 그룹 반복 시간: 알림 반복 시간:

일시 정지

시작일자 스케줄

스케줄: 지속시간:

수신자 목록

항목	설명
알림 수준	알림 경고 수준
알림 주기	알림 발송을 위한 시간을 설정 <ul style="list-style-type: none"> • 그룹 대기 시간: 처음 알림이 대기하는 시간 • 그룹 반복 시간: 알림 발송 후 재발송되기까지 대기 시간 • 알람 반복 시간: 알림 재발송 시간 간격

항목	설명
일시 정지	시작일자 혹은 스케줄 을 기준으로 지속 시간 만큼 알림 발송을 제한
수신자 목록	알림을 받을 클러스터 수신자 목록 (클러스터 수신자 메뉴에서 생성)

알림 규칙

알림 정책에 알림 규칙 추가 버튼으로 복수의 알림 규칙을 생성 할 수 있으며 알림 규칙 삭제 버튼으로 알림 규칙을 제거할 수 있다.

항목	설명
알림 규칙 이름	알림 규칙 이름
알림 규칙 설명	알림 규칙에 대한 설명
알림 대상	대상 지정 혹은 Query (Prometheus Query)
알림 수준	알림 경보 수준
알림 주기	알림 발송을 위한 시간을 설정 <ul style="list-style-type: none"> 그룹 대기 시간: 처음 알림이 대기하는 시간 그룹 반복 시간: 알림 발송 후 재발송되기까지 대기 시간 알람 반복 시간: 알림 재발송 시간 간격
일시 정지	시작일자 혹은 스케줄 을 기준으로 지속 시간 만큼 알림 발송을 제한

- 알림 대상이 대상 지정 의 경우

알림 규칙

알림 규칙

알림 규칙 이름

알림 규칙 설명

알림 대상

대상 지정 Query

오브젝트

단일 라벨 선택터

종류

네임스페이스

이름

메트릭 타입

임계 조건

%

알림 수준

알림 주기

일시 정지

알림 규칙 삭제

알림 규칙

알림 규칙

알림 규칙 이름

알림 규칙 설명

알림 대상

대상 지정 Query

오브젝트

단일 라벨 선택

종류

네임스페이스

라벨 선택

=
✕

메트릭 타입

임계 조건

알림 수준

알림 주기

일시 정지

오브젝트	단일 혹은 라벨 선택터 선택
종류	노드 혹은 워크로드(디플로이먼트, 스테이트풀셋, 데몬셋, 잡, 크론잡, 파드) 중 선택
네임스페이스	네임스페이스 선택
이름	오브젝트가 단일 의 경우 리소스 이름 선택
라벨 선택터	오브젝트가 라벨 선택터 의 경우 라벨 선택터 설정 <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <ul style="list-style-type: none"> 키: 매칭 할 라벨 키 값 밸류: 매칭 할 라벨 밸류 값 </div>

오브젝트	단일 혹은 라벨 셀렉터 선택
메트릭 타입	<ul style="list-style-type: none"> • 노드 메트릭 타입 <div style="border: 1px solid #ccc; padding: 10px; margin: 5px 0;"> <ul style="list-style-type: none"> • CPU 사용률 • 메모리 사용률 • 디스크 사용률 </div> • 워크로드 메트릭 타입 <div style="border: 1px solid #ccc; padding: 10px; margin: 5px 0;"> <ul style="list-style-type: none"> • CPU 사용률 • 메모리 사용률 • 액티브 서비스 • 응답시간 • Gc 시간 </div>
임계 조건	<ul style="list-style-type: none"> • 일치 • 불일치 • 보다 크다 • 크거나 같다 • 보다 작다 • 작거나 같다

오브젝트	단일 혹은 라벨 셀렉터 선택
	<p>임계 조건의 값 입력</p> <ul style="list-style-type: none"> • % (Percent) <ul style="list-style-type: none"> • CPU 사용률 • 메모리 사용률 • 디스크 사용률 • 개수 (Count) <ul style="list-style-type: none"> • 액티브 서비스 • 초 (Second) <ul style="list-style-type: none"> • 응답시간 • Gc 시간

- 알림 대상이 Query 의 경우

알림 규칙

알림 규칙

알림 규칙 이름

알림 규칙 설명

알림 대상

대상 지정
 Query

Query

임계 조건

임계 조건을 선택하세요.

설정값을 입력하세요.

알림 수준
 알림 주기
 일시 정지

알림 규칙 삭제

알림 규칙 추가

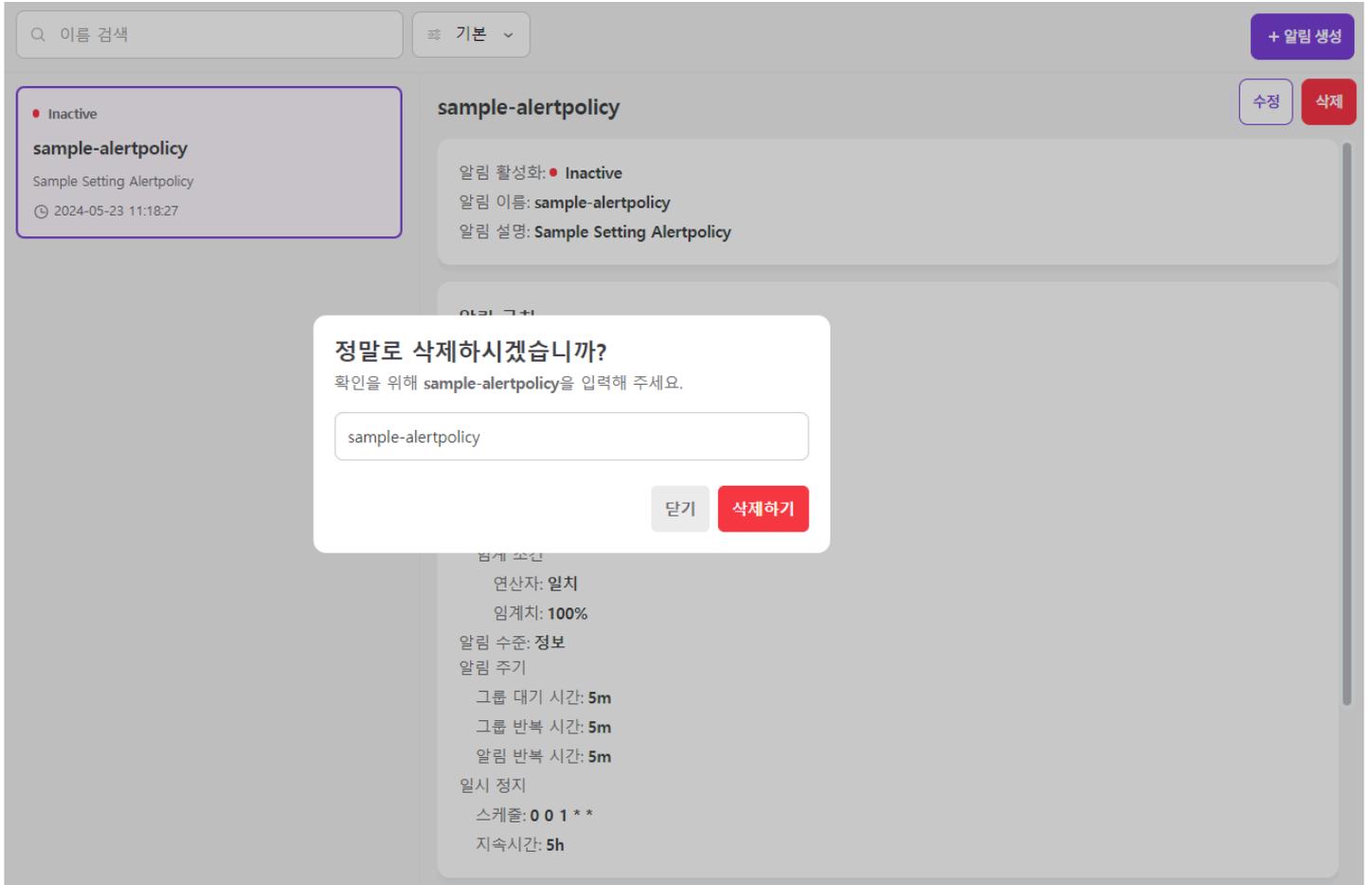
Query	Prometheus Query 입력
임계 조건	<ul style="list-style-type: none"> 일치 불일치 보다 크다 크거나 같다 보다 작다 작거나 같다
	임계 조건의 값에 해당하는 query 혹은 숫자 입력

알림 정책 수정

수정하려는 알림 정책을 선택하고 수정 버튼을 선택한다. 변경할 내용을 입력 또는 선택한 후 알림 수정 버튼을 눌러 반영한다.

알림 정책 삭제

삭제하려는 알림 정책을 선택하고 삭제 버튼을 선택한다.



모달에서 알림 정책 이름을 입력하여 삭제한다.

4.2.13.4. 레지스트리

레지스트리는 클러스터에서 공통으로 사용할 컨테이너 이미지 저장소를 관리한다.
 이 저장소는 클러스터에 배포된 네임스페이스에서 접근할 수 있다.
 아코디언에서는 기본으로 인프라 레지스트리와 사용자 레지스트리를 제공한다.

TIP

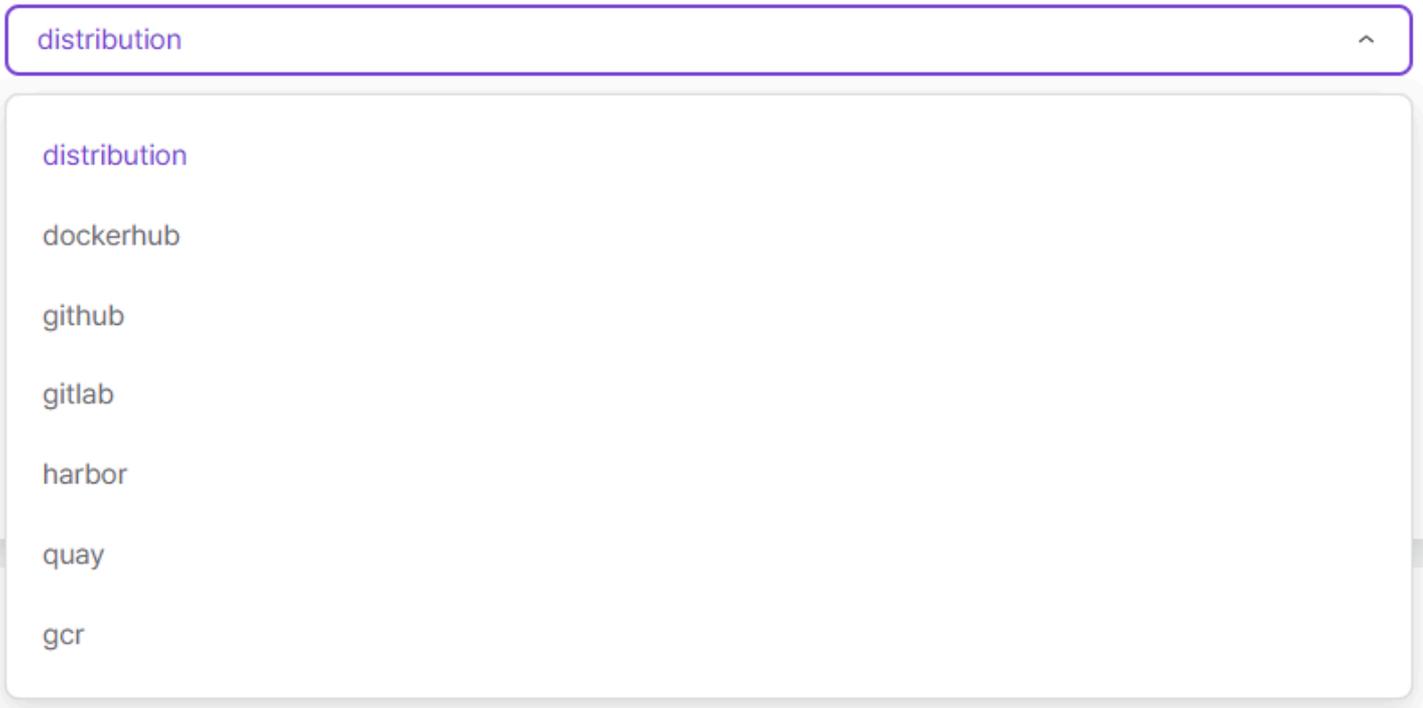
아코디언에서는 기본으로 인프라 레지스트리와 사용자 레지스트리를 제공한다.

인프라 레지스트리는 아코디언 구동에 필요한 인프라 컨테이너 이미지들을 저장하는 용도이다.
 사용자는 사용자 레지스트리만 사용한다.

사설인증서 (private certificate)로 서비스하는 컨테이너 레지스트리인 경우,
 컨테이너 런타임에서 해당 레지스트리의 SSL 인증서를 신뢰할 수 있도록 사전에 설정해야 한다.
 설정 방법은 사용하고 있는 컨테이너 런타임 매뉴얼을 참고한다.

추가적으로 사용자가 외부 및 내부 레지스트리를 정의하여 사용할 수 있다.
 아코디언에서 제공하는 레지스트리 벤더 종류 및 사용 가능한 관리 기능은 다음과 같다.

벤더



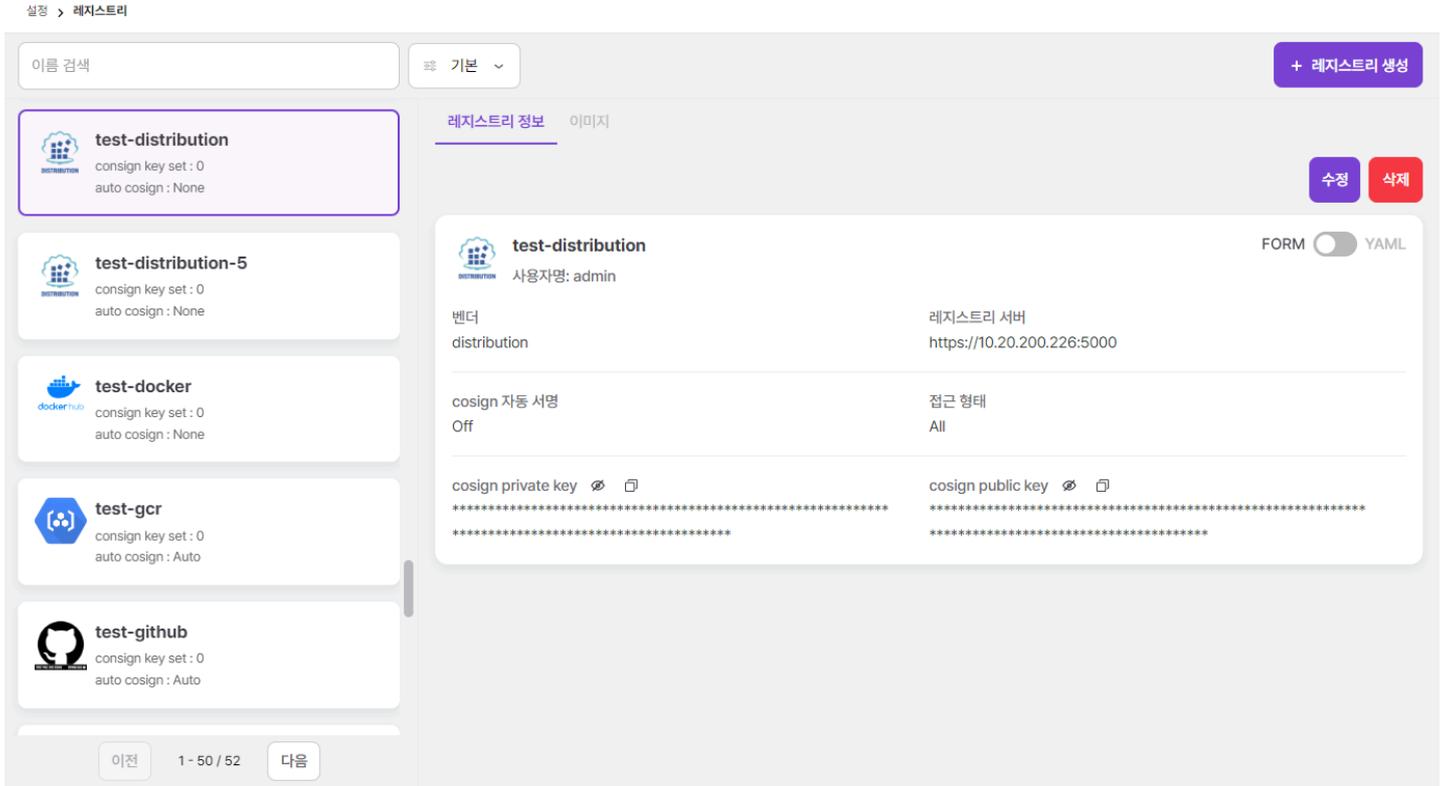
항목	(이미지, 태그)목록 조회	상세 조회	Cosign 서명	삭제
distribution	O	O	O	O
dockerhub	O	O	X	X
github	O	O	O	X
gitlab	O	O	O	O
harbor	O	O	O	O
quay	O	O	O	O

항목	(이미지, 태그)목록 조회	상세 조회	Cosign 서명	삭제
gcr	O	O	O	X

레지스트리 목록

레지스트리 메뉴 좌측에는 생성한 레지스트리들의 목록을 표시한다.

카드 형식으로 레지스트리의 이름과 코사인 자동서명 여부 및 추가로 등록된 검증 키의 개수를 확인할 수 있다.



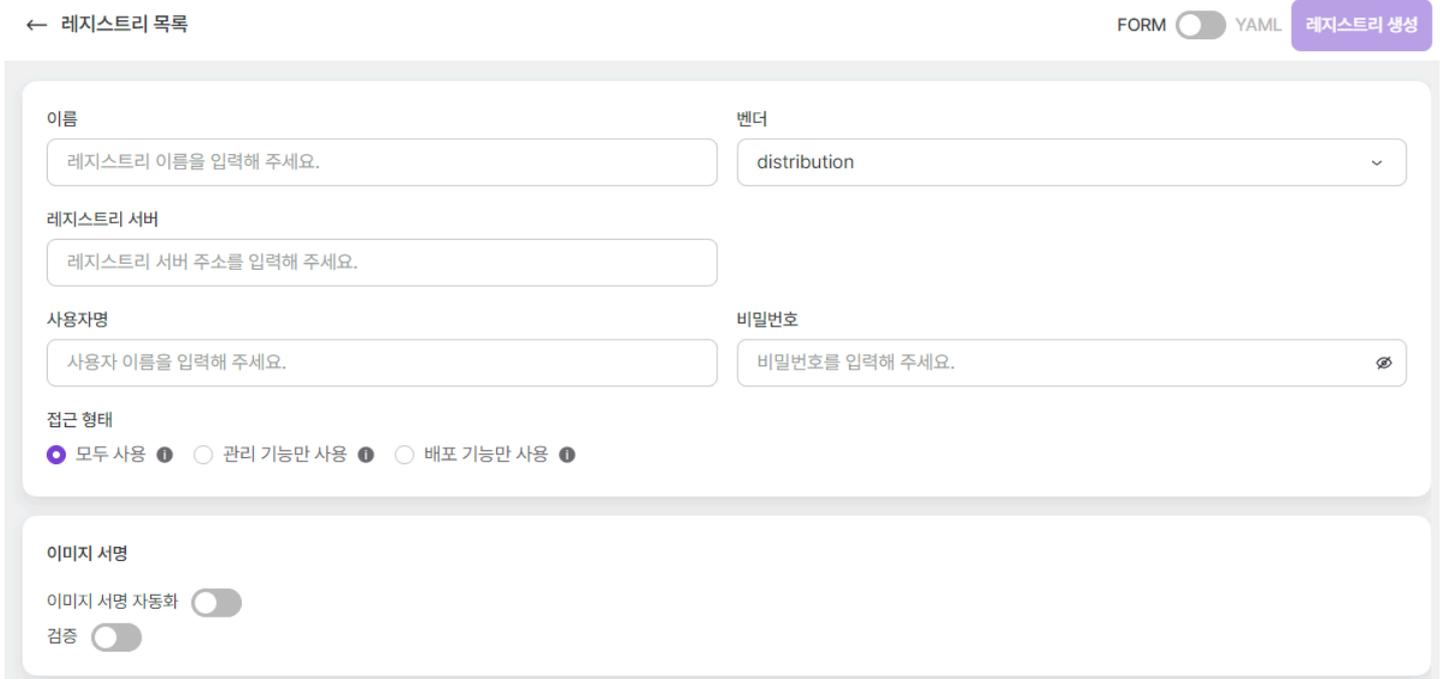
레지스트리 생성

+ 레지스트리 생성 을 선택하면 레지스트리 생성 페이지로 이동한다.

FORM 형태로 등록하는 방법과 YAML 편집기를 통해 등록하는 방법을 지원한다.

레지스트리 벤더 종류마다 입력해야하는 FORM이 다르다.

레지스트리에서 정보와 관련된 스펙은 모두 base64 인코딩된 상태로 입력 및 저장된다.



벤더별 레지스트리 생성 (FORM)

Distribution FORM

이름	벤더
레지스트리 이름을 입력해 주세요.	distribution
레지스트리 서버	
레지스트리 서버 주소를 입력해 주세요.	
사용자명	비밀번호
사용자 이름을 입력해 주세요.	비밀번호를 입력해 주세요.
접근 형태	
<input checked="" type="radio"/> 모두 사용 ⓘ	<input type="radio"/> 관리 기능만 사용 ⓘ <input type="radio"/> 배포 기능만 사용 ⓘ

항목	설명
이름	레지스트리 이름 (필수)
레지스트리 서버	레지스트리 서버 주소로 Http 와 Https를 구분한다. (필수)
사용자명	레지스트리 사용자 이름 (필수)
비밀번호	비밀번호 (필수)

TIP

레지스트리 서버 입력 예는 다음과 같다.

Ex) `http://registry.internal.co.kr`

배포 모드 사용 시, 파이프라인/카탈로그 빌드 시, 이미지 리포지터리에 사용자명이 자동 바인딩되어 푸시된다.

DockerHub FORM

이름	레지스트리 이름을 입력해 주세요.	벤더	dockerhub
사용자명	사용자 이름을 입력해 주세요.	액세스 토큰	액세스 토큰을 입력해 주세요.
접근 형태	<input checked="" type="radio"/> 모두 사용 ⓘ <input type="radio"/> 관리 기능만 사용 ⓘ <input type="radio"/> 배포 기능만 사용 ⓘ		
배포용 토큰	<input checked="" type="checkbox"/> ⓘ		

항목	설명
이름	레지스트리 이름 (필수)
사용자명	레지스트리 사용자 이름 (필수)
액세스 토큰	DockerHub에서 발급한 개인 액세스 토큰 (필수)

TIP

아코디언 레지스트리에서는 DockerHub를 사용할 경우, 개인 사용자의 인증 권한에 대해서만 지원한다.

액세스 토큰 발급 방법은 [다음](https://docs.docker.com/security/for-developers/access-tokens) (<https://docs.docker.com/security/for-developers/access-tokens>)을 확인한다.

GitHub FORM

이름

벤더

URL ⓘ

사용자명

액세스 토큰

조직 사용

접근 형태

모두 사용 ⓘ
 관리 기능만 사용 ⓘ
 배포 기능만 사용 ⓘ

배포용 토큰 ⓘ

항목	설명
이름	레지스트리 이름 (필수)
URL	레지스트리 서버의 URL로 리포지터리에서 이미지 이름 이전 경로까지인 사용자 이름 또는 그룹이름까지만 입력한다.
사용자명	레지스트리 토큰 사용자 이름 (필수)
액세스 토큰	GitHub에서 발급한 개인 액세스 토큰 (필수)
조직 사용	만약 레지스트리가 조직을 사용할 경우 체크한다.

TIP

URL 입력 예는 다음과 같다. Ex) acc , ghcr . io/acc

GitHub는 URL을 입력하지 않을 경우, 사용자명으로 자동 바인딩된다. Ex) ghcr . io/사용자명

액세스 토큰 발급 방법은 [다음](#)

(<https://docs.github.com/ko/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens>)

을 확인한다.

Gitlab FORM

이름

벤더

URL ⓘ

사용자명

액세스 토큰

접근 형태

모두 사용 ⓘ
 관리 기능만 사용 ⓘ
 배포 기능만 사용 ⓘ

배포용 토큰 ⓘ

항목	설명
이름	레지스트리 이름 (필수)
URL	관리 및 배포할 레지스트리 서버의 URL로 리포지터리에서 사용자명(또는 조직명)과 프로젝트 명까지 입력한다.
사용자명	레지스트리 토큰 사용자 이름 (필수)
액세스 토큰	Gitlab에서 발급한 개인 액세스 토큰 (필수)

TIP

URL 입력은 (사용자명 or 조직명)/(프로젝트명)으로 입력한다.

Ex) acc/test , registry.gitlab.com/acc/test

Gitlab은 사용자명(또는 조직명)과 프로젝트 이름을 같이 입력해야만 레지스트리 이미지를 정상적으로 조회 가능하다.

액세스 토큰 발급 방법은 [다음](https://docs.gitlab.com/ee/user/profile/personal_access_tokens.html) (https://docs.gitlab.com/ee/user/profile/personal_access_tokens.html)을 확인한다.

Harbor FORM

이름 레지스트리 이름을 입력해 주세요.	벤더 harbor
레지스트리 서버 레지스트리 서버 주소를 입력해 주세요.	
사용자명 사용자 이름을 입력해 주세요.	비밀번호 비밀번호를 입력해 주세요.
접근 형태 <input checked="" type="radio"/> 모두 사용 ⓘ <input type="radio"/> 관리 기능만 사용 ⓘ <input type="radio"/> 배포 기능만 사용 ⓘ	
배포용 토큰 ⓘ <input type="checkbox"/>	

항목	설명
이름	레지스트리 이름 (필수)
레지스트리 서버	레지스트리 서버 주소로 Http 와 Https를 구분한다. (필수)
사용자명	레지스트리 사용자 이름 (필수)
비밀번호	비밀번호 (필수)

TIP

레지스트리 서버 입력 예는 다음과 같다.

Ex) http://registry.internal.co.kr , http://registry.internal.co.kr/acc

배포 모드를 사용 시, 선택적으로 배포할 프로젝트 스코프를 직접 지정할 수 있다. 만약 프로젝트 스코프까지 입력하지 않을 경우, 시크릿 인증 정보를 생성할 때 프로젝트 스코프는 사용자명으로 자동 바인딩된다.

Ex)

http://registry.internal.co.kr → http://registry.internal.co.kr/사용자명

http://registry.internal.co.kr/acc → http://registry.internal.co.kr/acc

Quay FORM

이름	벤더
레지스트리 이름을 입력해 주세요.	quay
사용자명	비밀번호
사용자 이름을 입력해 주세요.	비밀번호를 입력해 주세요.
조직 사용 <input type="checkbox"/>	
접근 형태	
<input checked="" type="radio"/> 모두 사용 ⓘ <input type="radio"/> 관리 기능만 사용 ⓘ <input type="radio"/> 배포 기능만 사용 ⓘ	
배포용 토큰 ⓘ <input type="checkbox"/>	

항목	설명
이름	레지스트리 이름 (필수)
사용자명	레지스트리 사용자 이름 (필수)
비밀번호	Quay에서 설정한 도커 로그인 비밀번호 (필수)
조직 사용	만약 레지스트리가 조직을 사용할 경우 체크한 뒤 조직 이름을 입력한다.

TIP

Quay는 파이프라인/카탈로그 빌드 시, 이미지의 리포지터리에 기본적으로 사용자명 또는 조직명이 자동 바인딩된다. Ex) quay.io/(사용자명 or 조직명)

Gcr FORM

이름

벤더

URL ⓘ

서비스 계정 키

접근 형태

모두 사용 ⓘ
 관리 기능만 사용 ⓘ
 배포 기능만 사용 ⓘ

배포용 토큰 ⓘ

항목	설명
이름	레지스트리 이름 (필수)
URL	레지스트리 서버의 URL로 컨테이너 레지스트리의 경우 프로젝트이름, 아티팩트 저장소인 경우 스토리지 저장소까지 입력한다.
서비스 계정 키	GCP에서 발급한 개인 액세스 토큰 (필수)

TIP

컨테이너 레지스트리를 사용하고 있는 경우, (region).gcr.io 도메인을 사용해야한다. URL에서 해당 도메인을 포함하여 프로젝트ID까지 입력해야한다. Ex) gcr.io/accordion-0123

만약 아티팩트 레지스트리를 사용하고 있다면, (region).gcr.io를 사용하거나 (region)-docker.pkg.dev 도메인을 사용한다. URL은 도메인을 포함하여 프로젝트 ID를 포함한 스토리지 저장소까지 입력해야한다.

아티팩트 레지스트리는 스토리지 저장소가 설정되어 있지 않다면 레지스트리를 이용할 수 없다.

서비스 계정 권한에 대한 설정 방법은 다음을 확인한다.

- [컨테이너 레지스트리](https://cloud.google.com/container-registry/docs/access-control?authuser=1&hl=ko) (https://cloud.google.com/container-registry/docs/access-control?authuser=1&hl=ko)
- [아티팩트 레지스트리](https://cloud.google.com/artifact-registry/docs/access-control?authuser=1&hl=ko) (https://cloud.google.com/artifact-registry/docs/access-control?authuser=1&hl=ko)

서비스 계정 키에 대한 발급 방법은 다음을 확인한다.

- [컨테이너 레지스트리](https://cloud.google.com/container-registry/docs/advanced-authentication?hl=ko#json-key) (https://cloud.google.com/container-registry/docs/advanced-authentication?hl=ko#json-key)
- [아티팩트 레지스트리](https://cloud.google.com/artifact-registry/docs/docker/authentication?hl=ko&authuser=1#json-key) (https://cloud.google.com/artifact-registry/docs/docker/authentication?hl=ko&authuser=1#json-key)

다음은 공통적으로 존재하는 FORM이다.

접근 형태

모두 사용 ⓘ
 관리 기능만 사용 ⓘ
 배포 기능만 사용 ⓘ

배포용 토큰 ⓘ

이미지 서명

이미지 서명 자동화

AUTO GENERATE KEY

Cosign public key

Cosign Public Key를 입력해 주세요.

Cosign private key

Cosign Private Key를 입력해 주세요.

검증

Key 추가

항목	설명
접근 형태	레지스트리 이용 용도에 따라 사용할 모드를 선택한다. <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <ul style="list-style-type: none"> 모두 사용 : 관리와 배포 기능을 모두 사용한다. 관리 기능만 사용 : 컨테이너 이미지 저장소 내, 이미지 조회나 삭제와 같은 이미지 관리 기능만 사용한다. 배포 기능만 사용 : 컨테이너 이미지 저장소 인증 정보 시크릿을 각 네임스페이스마다 배포하여 파이프라인/카탈로그에서 사용한다. </div>
배포용 토큰	접근 형태 모두 사용 시, 관리와 배포에 대한 인증 권한 정보를 분리하고 싶을 경우 사용한다. 해당 옵션 체크 시, 시크릿을 배포할 때 배포용 토큰 정보로 시크릿이 생성된다.

아코디언에서 벤더의 Cosign 이미지 서명 기능을 지원할 경우, 다음과 같이 추가적인 옵션을 사용할 수 있다.

항목	설명
Cosign 자동 서명	해당 옵션을 체크할 경우, 파이프라인/카탈로그 빌드 시, 이미지 서명도 함께 진행한다.
AUTO GENERATE KEY	Cosign 자동 서명을 사용할 경우에만 사용 가능하며, Cosign 서명키가 따로 존재하지 않을 경우, 체크하여 해당 레지스트리의 서명 키를 생성한다.

항목	설명
Cosign public/private key	Cosign 자동 서명을 사용할 경우에만 사용 가능하며, Cosign 서명키가 따로 존재할 경우, 직접 입력한다.
외부에서 서명된 컨테이너 이미지 검증	Cosign 이미지 서명에 대한 검증 기능을 사용할 때, 등록된 키로 서명 검증을 할 수 있다.

레지스트리 생성 (YAML)

← 레지스트리 목록

FORM YAML 레지스트리 생성

```

1  apiVersion: containermanagement.accordions.co.kr/v1beta1
2  kind: Registry
3  metadata:
4    name: ''
5  spec:
6    activeMode: All
7    additional:
8      cosign:
9        signPolicy: None
10   provider:
11     distribution:
12       server: ''
13       username: ''
14       password: ''
15

```

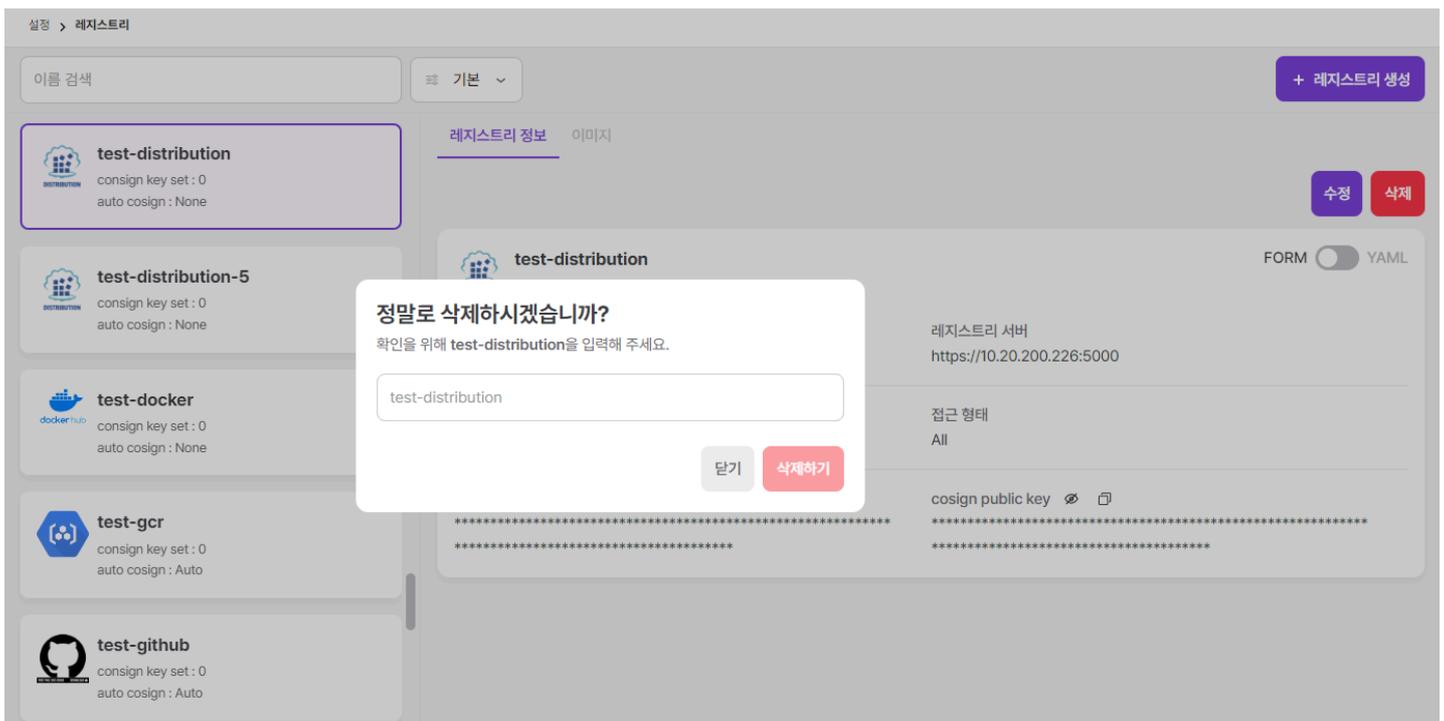
YAML 편집기로도 레지스트리 생성이 가능하다. YAML 편집기로 생성할 경우, 벤더의 스펙은 모두 base64 인코딩한 상태로 입력해야한다.

레지스트리 수정

수정하려는 레지스트리를 선택하고 우측의 수정 버튼을 선택하여 원하는 수정 항목을 반영한다. 한번 생성한 레지스트리의 레지스트리 명과 벤더 종류는 변경할 수 없다.

레지스트리 삭제

삭제하려는 레지스트리를 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 레지스트리 이름을 입력하여 삭제한다.

레지스트리 정보

레지스트리에 대한 설정 정보를 확인할 수 있다.

The screenshot shows a web interface for managing registries. On the left, there is a list of registries: 'copt-test-registry', 'd1', 'd10', and 'd11'. The 'copt-test-registry' is selected and its details are shown on the right. The details include:

- copt-test-registry** (FORM YAML)
- 사용자명: accregistry
- 벤더: distribution (레지스트리 서버: 10.60.100.46:5000)
- cosign 자동 서명: On (접근 형태: All)
- cosign private key and cosign public key fields with masked content.

항목	설명
이름	레지스트리의 이름이 표시된다.
사용자명	레지스트리 인증 정보의 사용자명이 표시된다.
벤더	레지스트리 벤더 종류가 표시된다.
레지스트리 서버	레지스트리 서버가 표시된다. 일부 벤더의 경우 조회 및 배포에 필요한 리포지터리 스코프까지 표시된다.
cosign 자동 서명	Cosign 자동 서명 여부가 표시된다.
접근 형태	레지스트리 사용 모드가 표시된다.
cosign private key	Cosign 서명에 사용하는 private key가 표시된다.
cosign public key	Cosign 서명 검증에 사용하는 public key가 표시된다.

추가적으로 우측 상단에 YAML 토글 버튼을 클릭하면 YAML 스펙으로도 확인할 수 있다.

이름 검색

☰ 기본 ▾

+ 레지스트리 생성

copt-test-registry
consign key set : 1
auto cosign : Auto

d1
consign key set : 0
auto cosign : None

d10
consign key set : 0
auto cosign : None

d11
consign key set : 0
auto cosign : None

이전

1 - 50 / 52

다음

레지스트리 정보 이미지

수정

삭제

copt-test-registry
사용자명: accregistry

FORM
 YAML

```

1 kind: Registry
2 apiVersion: containermanagement.accordions.co.kr/v1beta1
3 metadata:
4   creationTimestamp: '2024-01-04T07:10:19Z'
5   generation: 7
6   managedFields: ...
51  name: copt-test-registry
52  resourceVersion: '55534907'
53  uid: f543ef8f-4961-4c7a-9dcd-ea12714dd5c9
54 spec:
55   activeMode: All
56   additional:
57     cosign:
58       signKey:
59         privateKey: >-
60         LS0tLS1CRUdJTiBFTkNSWVBUU01HU1RPUkUgUFJkFURSBkVktLS0tLQpleU9uc2

```

이미지 (리포지터리) 목록

레지스트리의 이미지 목록을 보여준다.

입력한 인증 권한에 맞는 이미지만 조회 가능하며, 검색 기능 사용 시 3글자 이상 입력해야 검색할 수 있다.

설정 > 레지스트리

기본
+ 레지스트리 생성

distribution
consign key set : 2
auto cosign : None

dockerhub
consign key set : 0
auto cosign : None

infra-registry
consign key set : 0
auto cosign : None

user-registry
consign key set : 0
auto cosign : None

레지스트리 정보
이미지

검색
total:116
업로드

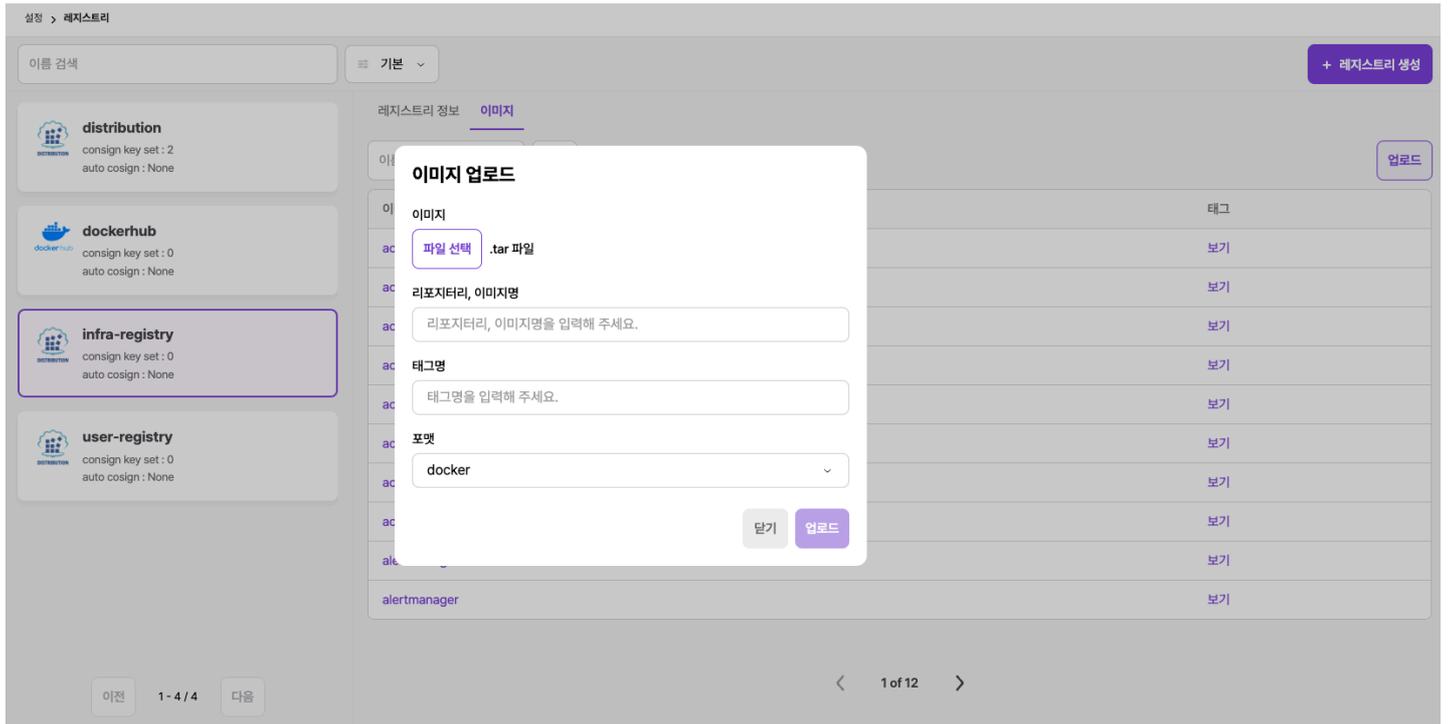
이미지명 ↑	태그
accordion/ant	보기
accordion/cicd-image-kaniko	보기
accordion/cicd-vcs-git	보기
accordion/cicd-vcs-svn	보기
accordion/openjdk	보기
accordion/tomcat	보기
accordion/wildfly	보기
accordion-package/console	보기
alertmanagement-server	보기
alertmanager	보기

이전
1 - 4 / 4
다음

항목	설명
total	조회한 레지스트리의 총 이미지 수가 표시된다.
이미지명	레지스트리의 이미지 저장소 이름이 표시된다.
태그	보기 버튼을 선택하면 레지스트리 이미지 저장소가 가지고 있는 총 태그의 수가 표시된다.
업로드	레지스트리 내 이미지를 업로드 할 수 있는 모달이 출력된다.

이미지 업로드

레지스트리 내 이미지 업로드를 지원한다.



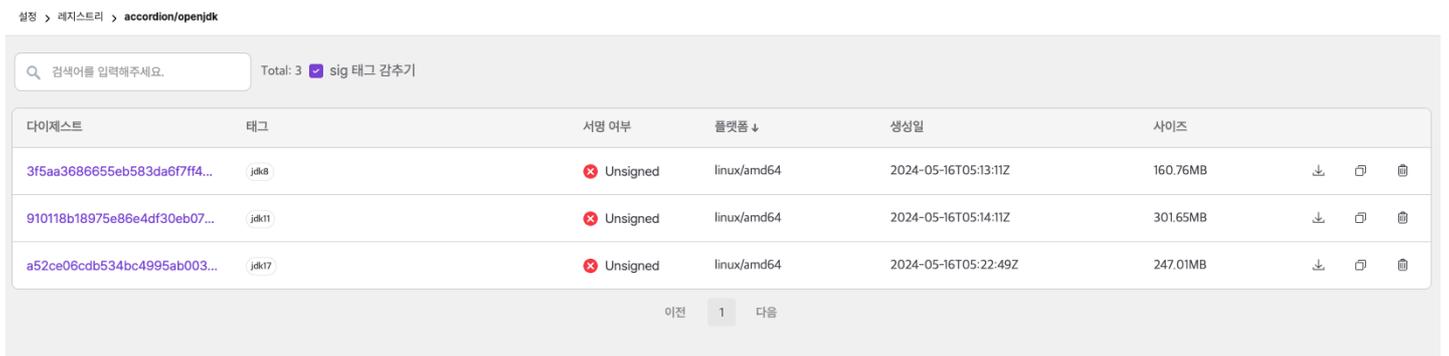
항목	설명
이미지	업로드 할 이미지(확장자는 .tar)
리포지터리, 이미지명	업로드 할 리포지터리, 이미지명
태그명	업로드할 이미지에 붙일 태그명
포맷	업로드할 이미지의 포맷(docker 와 oci 지원) 업로드한 파일 포맷이 설정한 포맷과 다른 경우 업로드 실패 docker의 경우 멀티 이미지 업로드를 지원하지 않음

이미지 다이제스트 목록

리포지터리에서 사용되는 다이제스트를 목록을 보여준다.

sig 태그 감추기 옵션을 통해 순수한 이미지 다이제스트만 목록을 호출할 수 있다.

(sig 태그 란 Cosign 시 생성되는 태그다.)



항목	설명
다이제스트	다이제스트명
태그	다이제스트가 가지고 있는 태그의 목록
서명 여부	다이제스트의 Cosign 상태를 나타낸다. signed: 서명된 상태 unsigned: 미서명된 상태
플랫폼	manifest 의 플랫폼 정보가 표시된다. indexManifests 의 경우 모든 manifest 의 플랫폼 정보가 표시된다.
생성일	다이제스트의 생성일. indexManifests 의 경우 N/A 로 표시된다.
사이즈	다이제스트의 사이즈. indexManifests 의 경우 N/A 로 표시된다.
	<div data-bbox="454 810 1497 1576" style="border: 1px solid #ccc; padding: 10px;"> <p style="font-size: 1.2em; font-weight: bold; margin-bottom: 10px;">sha256:a52ce06cdb534bc4995ab00302fd4c...</p> <p>포맷</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px; display: flex; justify-content: space-between; align-items: center;"> docker ▼ </div> <p>플랫폼</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px; display: flex; justify-content: space-between; align-items: center;"> linux/amd64 ▼ </div> <p>태그명</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px; min-height: 20px;"> 태그명을 입력해 주세요. </div> <div style="display: flex; justify-content: flex-end; gap: 10px;"> 닫기 다운로드 </div> </div> <p style="font-size: 0.9em; margin-top: 10px;">다이제스트 이미지에 대해 다운로드를 지원한다. `docker` 와 `oci` 포맷을 지원하며, 해당 다이제스트의 플랫폼을 선택할 수 있다. 전체 플랫폼의 경우 `oci` 포맷 이미지만 지원한다. 태그명을 입력하지 않으면 다운로드시 다이제스트명으로 다운로드되며, 태그명을 입력할 시 해당 태그명으로 다운로드 된다.</p>

항목	설명
	<div data-bbox="454 179 1497 593"> <h3>서명하시겠습니까?</h3> <p>다이제스트 [sha256:dede560d9526b5abb996cb3149de165e12e946676d55808ce23ecde2c527cecd]를 서명합니다.</p> <p style="text-align: right;"> 닫기 서명하기 </p> </div> <p>DockerHub를 제외한 레지스트리 중 이미지 서명 자동화를 설정한 경우 Cosign을 지원한다. 레지스트리가 가지고 있는 privateKey 를 이용하여 이미지 다이제스트를 Cosign 서명한다. 만약 indexManifests인 경우 recursive 하게 하위의 다이제스트들도 모두 서명 처리된다. 서명 처리된 다이제스트의 경우, 레지스트리가 가지고 있는 publicKey 또는 verifyKeys 를 이용하여 검증할 수 있다.</p>
	<div data-bbox="454 835 1497 1534"> <h3>태그 복사하기</h3> <p>다이제스트명</p> <p>sha256:dede560d9526b5abb996cb3149de165e12e946676d55808ce23</p> <p>레지스트리</p> <p>레지스트리를 선택해주세요. ▾</p> <p>이미지명 변경 : 태그명 변경</p> <p>새 이미지명을 입력해 주세요. : 새 태그명을 입력해 주세요.</p> <p><input type="checkbox"/> 태그 덮어쓰기</p> <p style="text-align: right;"> 닫기 복사하기 </p> </div> <p>다이제스트를 복사한다. 복사할 대상의 레지스트리를 선택하고 이미지명, 태그명을 변경할 수 있다. 태그명을 변경하지 않을 경우 기본적으로 latest 태그로 복사된다. 태그 덮어쓰기 를 선택한 경우 기존에 동일한 태그명이 존재할 경우 무시하고 덮어씌워진다.</p>

항목	설명
	<div data-bbox="456 181 1495 725" style="border: 1px solid #ccc; padding: 10px;"> <p>정말로 삭제하시겠습니까?</p> <p>확인을 위해 sha256:dede560d9526b5abb996cb3149de165e12e946676d55808ce23ecde2c527cecd을 입력해 주세요.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin: 10px 0;"> sha256:dede560d9526b5abb996cb3149de165e12e94 </div> <div style="display: flex; justify-content: flex-end; gap: 10px;"> 닫기 삭제하기 </div> </div> <p data-bbox="475 781 1102 837">DockerHub, GitHub, Gcr을 제외한 벤더인 경우 삭제를 지원하지 않습니다. 삭제하기를 선택할 경우 다이제스트를 삭제합니다.</p>

문제가 있는 다이제스트의 경우

다이제스트 내 문제가 있는 경우 경고 아이콘을 출력한다. 해당 다이제스트는 삭제를 제외한 상세 정보 조회 및 복사 등의 추가적인 기능을 지원하지 않는다. 삭제를 지원하지 않는 프로바이더의 또는 삭제 실패시, 아코디언에서는 삭제 처리가 불가하여 해당 프로바이더에서 삭제가 필요하다.

설정 > 레지스트리 > console

검색어를 입력해주세요. Total: 178 sig 태그 감추기

다이제스트	태그	서명 여부	플랫폼 ↓	생성일	사이즈	
d3ae5e72cd9740fdc4d3c8ba...	ef038f99c61041fb3f1f289d57795c2c106f503318f0f5990c68656b	✖ Unsigned		2023-04-06T06:30:06Z	20.76MB	📄 ⬇️ 🗑️
66fed5a35daede3e4893fad2...	f3c26a4f99e6bf092586b3d97d15ff99a77ba1842b8ba0af97b89cd	✖ Unsigned		2023-04-10T06:07:54Z	20.71MB	📄 ⬇️ 🗑️
⚠️ ba1de5f5995ffe55195943...	2.6.0	✖ Unsigned				📄 ⬇️ 🗑️
663ee2052bf1b3fd30d94f4d3...	test-memory	✖ Unsigned	linux/amd64	2024-05-14T01:22:29Z	103.96MB	📄 ⬇️ 🗑️
ba5315da0b7ab42a578130c2...	2.7.1	✖ Unsigned	linux/amd64	2024-06-04T08:50:02Z	104.27MB	📄 ⬇️ 🗑️
flcc077388e8e1da1d0b9b433...	apn-test2	✖ Unsigned	linux/amd64	2024-07-26T05:13:11Z	95.49MB	📄 ⬇️ 🗑️
a409beef192198798a4a16983...	2.4.0	✖ Unsigned	linux/amd64	2023-05-10T07:33:54Z	32.21MB	📄 ⬇️ 🗑️
e6f05054dd1b7b3eaa5737f7c...	2.2.5	✖ Unsigned	linux/amd64	2022-11-14T00:17:14Z	35.59MB	📄 ⬇️ 🗑️
b21a62010530c9e1ff4095f31a...	e7781cc227c04f602011df28e00d154d06e98a9b4293a67544f9ff	✖ Unsigned		2023-03-08T23:53:04Z	691.62MB	📄 ⬇️ 🗑️
1d004a1d746099d30cddea67...	134bb256c9d1e2e363197969b8cf75d134113d19a594eba9de49b7	✖ Unsigned		2023-03-09T00:05:28Z	19.90MB	📄 ⬇️ 🗑️

이전 1 ... 4 5 6 7 8 ... 18 다음

이미지 다이제스트 정보

다이제스트 정보를 manifest 기반으로 보여준다.

설정 > 레지스트리 > admin/cicd-image-kaniko > sha256:262e0dc6b33c62099bc0be76a96cc5397007a572b3bc3cdeeebbd24bae27b7e3

✖ unsigned

10.20.200.206:30001/admin/cicd-image-kaniko@sha256:262e0dc6b33c62099bc0be76a96cc5397007a572b3bc3cdeeebbd24bae27b7e3

📄 ⋮

플랫폼

생성일

사이즈

linux/amd64

2022-01-10T12:15:06Z

43.51MB

Config

Layers

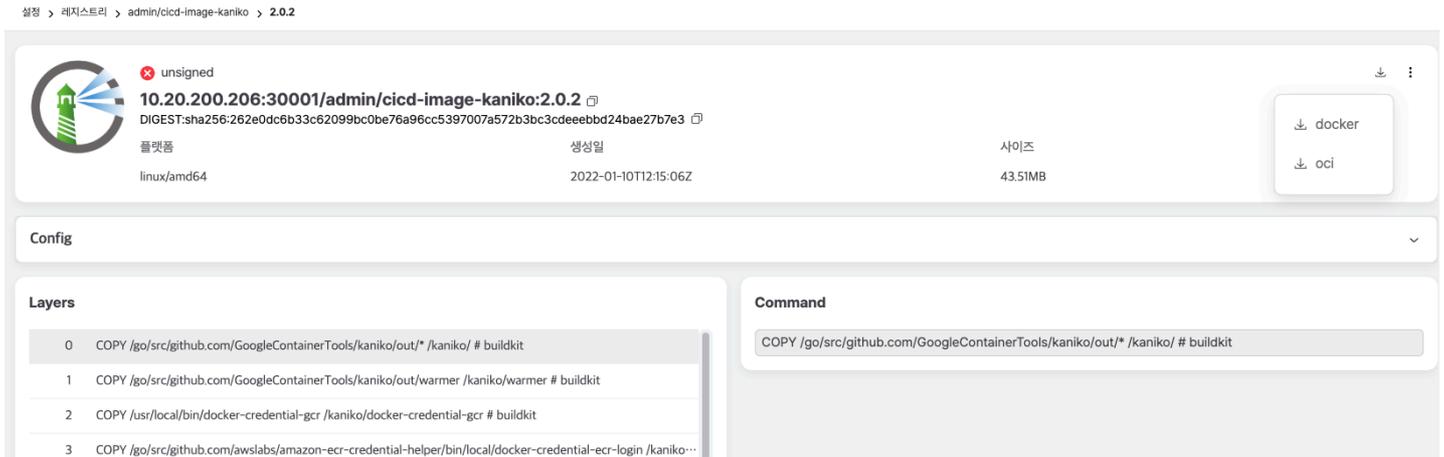
- 0 COPY /go/src/github.com/GoogleContainerTools/kaniko/out/* /kaniko/ # buildkit
- 1 COPY /go/src/github.com/GoogleContainerTools/kaniko/out/warmer /kaniko/warmer # buildkit
- 2 COPY /usr/local/bin/docker-credential-gcr /kaniko/docker-credential-gcr # buildkit
- 3 COPY /go/src/github.com/awslabs/amazon-ecr-credential-helper/bin/local/docker-credential-ecr-login /kaniko...
- 4 COPY /go/src/github.com/chrisrnellard/docker-credential-acr-env/build/docker-credential-acr-env /kaniko/doc...
- 5 COPY /bin /busybox # buildkit
- 6 VOLUME [/busybox]
- 7 COPY /ca-certificates.crt /kaniko/ssl/certs/ # buildkit
- 8 COPY /kaniko/.docker /kaniko/.docker # buildkit
- 9 COPY files/nsswitch.conf /etc/nsswitch.conf # buildkit
- 10 ENV HOME=/root
- 11 ENV USER=root

Command

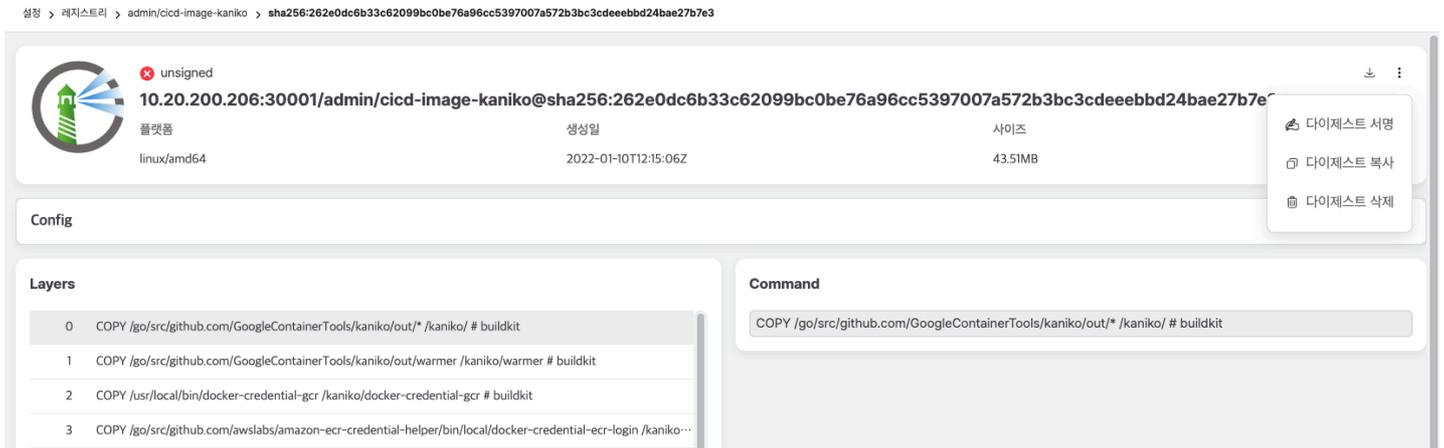
```
COPY /go/src/github.com/GoogleContainerTools/kaniko/out/* /kaniko/ # buildkit
```

다이제스트 관련 동작 버튼

우측 상단의  을 누르면 다이제스트 이미지를 다운로드를 위한 포맷 셀렉트 박스가 나오며, 포맷 정보를 누르면 이미지가 다운로드 된다.



우측 상단의  을 누르면 다이제스트에 대한 서명, 복사, 삭제에 대한 동작을 할 수 있는 셀렉트 박스가 나오며, 다이제스트 목록의 동작과 동일한 방식으로 동작된다.



항목	설명
다이제스트 서명	다이제스트를 서명한다.
다이제스트 복사	다이제스트를 복사한다. 복사할 대상의 레지스트리를 선택하고 이미지명, 태그명을 변경할 수 있다. 태그명을 변경하지 않을 경우 기본적으로 latest 태그로 복사된다. 태그 덮어쓰기 를 선택한 경우 기존에 동일한 태그명이 존재할 경우 무시하고 덮어씌워진다.
다이제스트 삭제	다이제스트를 삭제한다.

Layers 및 Command

Layers의 Row를 선택하면 우측 Command에 선택한 Layer의 Command 전체 내용이 출력된다.

Layers

1	/bin/sh -c apt-get update && apt-get install -y --no-install-recom...
2	/bin/sh -c apt-get update && apt-get install -y --no-install-recom...
3	/bin/sh -c apt-get update && apt-get install -y --no-install-recom...
4	/bin/sh -c apt-get update && apt-get install -y --no-install-recom...
5	/bin/sh -c echo 'deb http://deb.debian.org/debian jessie-backport...
6	ENV LANG=C.UTF-8
7	/bin/sh -c { echo '#!/bin/sh'; echo 'set -e'; echo; echo 'dirname "\${(r...
8	ENV JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
9	ENV JAVA_VERSION=8u111
10	ENV JAVA_DEBIAN_VERSION=8u111-b14-2-bpo8+1
11	ENV CA_CERTIFICATES_JAVA_VERSION=20140324
12	/bin/sh -c set -x && apt-get update && apt-get install -y openjdk-...
13	/bin/sh -c /var/lib/dpkg/info/ca-certificates-java.postinst configure

Command

```
/bin/sh -c { echo '#!/bin/sh'; echo 'set -e'; echo; echo 'dirname "${(r
eadlink -f "${which javac || which java}")}"; } > /usr/local/bin/docker-java-ho
me && chmod +x /usr/local/bin/docker-java-home
```

리포지터리 태그 정보

리포지터리 태그 정보는 다이제스트 정보와 거의 동일한 화면을 보여준다.
추가적으로 해당 태그의 digest 만 별도로 표시된다.

설정 > 레지스트리 > admin/cicd-image-kaniko > 2.0.2



10.20.200.206:30001/admin/cicd-image-kaniko:2.0.2

DIGEST:sha256:262e0dc6b33c62099bc0be76a96cc5397007a572b3bc3cdeeebbd24bae27b7e3

플랫폼: linux/amd64

unsigned

생성일: 2022-01-10T12:15:06Z

사이즈: 43.51MB

Layers

0	COPY /go/src/github.com/GoogleContainerTools/kaniko/out/* /kaniko/ # buildkit
1	COPY /go/src/github.com/GoogleContainerTools/kaniko/out/warmer /kaniko/warmer # buildkit
2	COPY /usr/local/bin/docker-credential-gcr /kaniko/docker-credential-gcr # buildkit
3	COPY /go/src/github.com/awslabs/amazon-ecr-credential-helper/bin/local/docker-credential-ecr-login /kaniko...
4	COPY /go/src/github.com/chrismlard/docker-credential-acr-env/build/docker-credential-acr-env /kaniko/doc...
5	COPY /bin /busybox # buildkit
6	VOLUME [/busybox]
7	COPY /ca-certificates.crt /kaniko/ssl/certs/ # buildkit
8	COPY /kaniko/docker /kaniko/docker # buildkit
9	COPY files/nsswitch.conf /etc/nsswitch.conf # buildkit
10	ENV HOME=/root

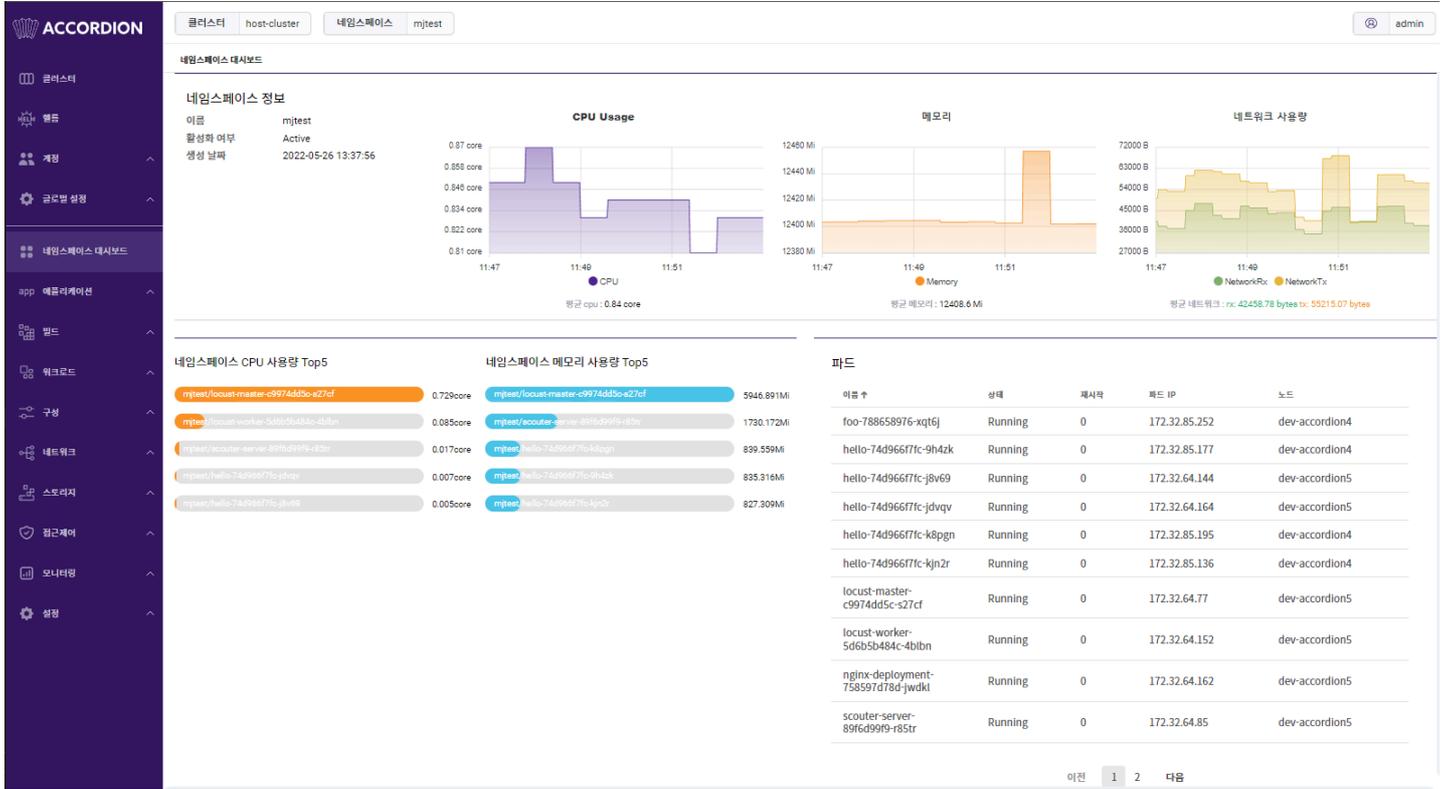
Command

```
COPY /go/src/github.com/GoogleContainerTools/kaniko/out/* /kaniko/ # buildkit
```


4.3. 네임스페이스 메뉴

4.3.1. 네임스페이스 대시보드

네임스페이스 대시보드는 네임스페이스 정보 및 네임스페이스 기준의 시스템 리소스(CPU/메모리 등), 네임스페이스에 배포되어있는 파드들의 상태 정보를 제공한다.



네임스페이스 대시보드 상단에서는 네임스페이스 설명 및 시스템 리소스 사용에 대한 정보를 제공한다.

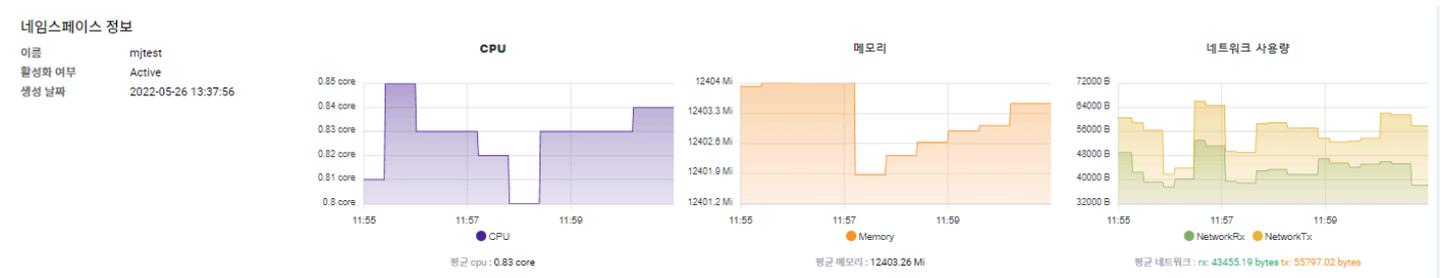


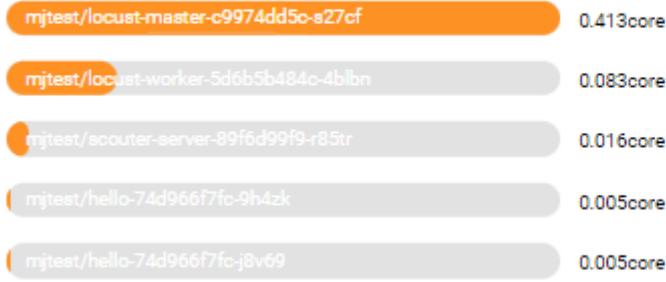
Table 10. 네임스페이스 정보

항목	설명
이름	네임스페이스 이름
활성화 여부	네임스페이스 상태 (Active 또는 Terminating) <ul style="list-style-type: none"> Active: 정상 배포 Terminating: 삭제 중
생성 날짜	생성된 시간
CPU	5분간 CPU 사용량 (평균 CPU: CPU의 평균값)

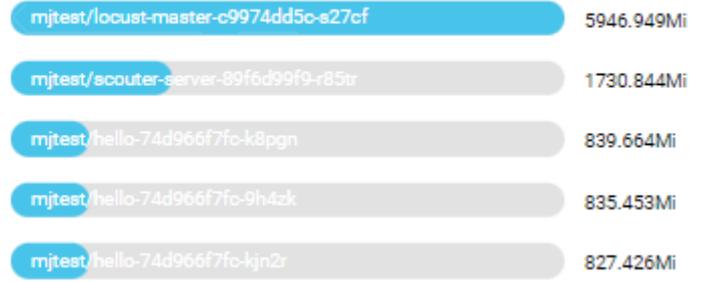
항목	설명
메모리	5분간 메모리 사용량 (평균 메모리: 메모리의 평균값)
네트워크 사용량	5분간 네트워크 사용량 (평균 네트워크: 네트워크 사용량의 평균값)

네임스페이스 대시보드 하단은 네임스페이스에 배포된 파드에 대한 정보를 제공한다. 좌측의 차트는 해당 네임스페이스에서 CPU 사용량이 가장 많은 파드와 메모리 사용량이 가장 많은 파드 각각 5개를 알려준다.

네임스페이스 CPU 사용량 Top5



네임스페이스 메모리 사용량 Top5



우측의 목록은 네임스페이스에 배포된 모든 파드의 목록을 나열한다.

파드

이름 ↑	상태	재시작	파드 IP	노드
foo-788658976-xqt6j	Running	0	172.32.85.252	dev-accordion4
hello-74d966f7fc-9h4zk	Running	0	172.32.85.177	dev-accordion4
hello-74d966f7fc-j8v69	Running	0	172.32.64.144	dev-accordion5
hello-74d966f7fc-jdvqv	Running	0	172.32.64.164	dev-accordion5
hello-74d966f7fc-k8pgn	Running	0	172.32.85.195	dev-accordion4
hello-74d966f7fc-kjn2r	Running	0	172.32.85.136	dev-accordion4
locust-master-c9974dd5c-s27cf	Running	0	172.32.64.77	dev-accordion5
locust-worker-5d6b5b484c-4blbn	Running	0	172.32.64.152	dev-accordion5
nginx-deployment-758597d78d-jwdkl	Running	0	172.32.64.162	dev-accordion5
scouter-server-89f6d99f9-r85tr	Running	0	172.32.64.85	dev-accordion5

항목	설명
이름	파드의 이름
상태	파드의 구동 상태
재시작	파드의 재시도 횟수
파드 IP	파드의 IP

항목	설명
노드	파드가 배포된 노드

4.3.2. 애플리케이션

네임스페이스 스코프에 있는 애플리케이션은 사용자가 헬름 차트 또는 카탈로그 템플릿을 이용해 애플리케이션을 배포할 수 있도록 기능을 제공한다. 그중 헬름 차트를 이용한 배포는 클러스터 스코프의 애플리케이션에 있는 헬름 메뉴의 기능과 스코프를 제외하고 동일하기 때문에 해당 가이드를 참고한다. 본 장에서는 카탈로그 메뉴에서 카탈로그 템플릿을 이용해 애플리케이션을 배포하고 관리하는 방법을 알아본다.

4.3.2.1. 카탈로그

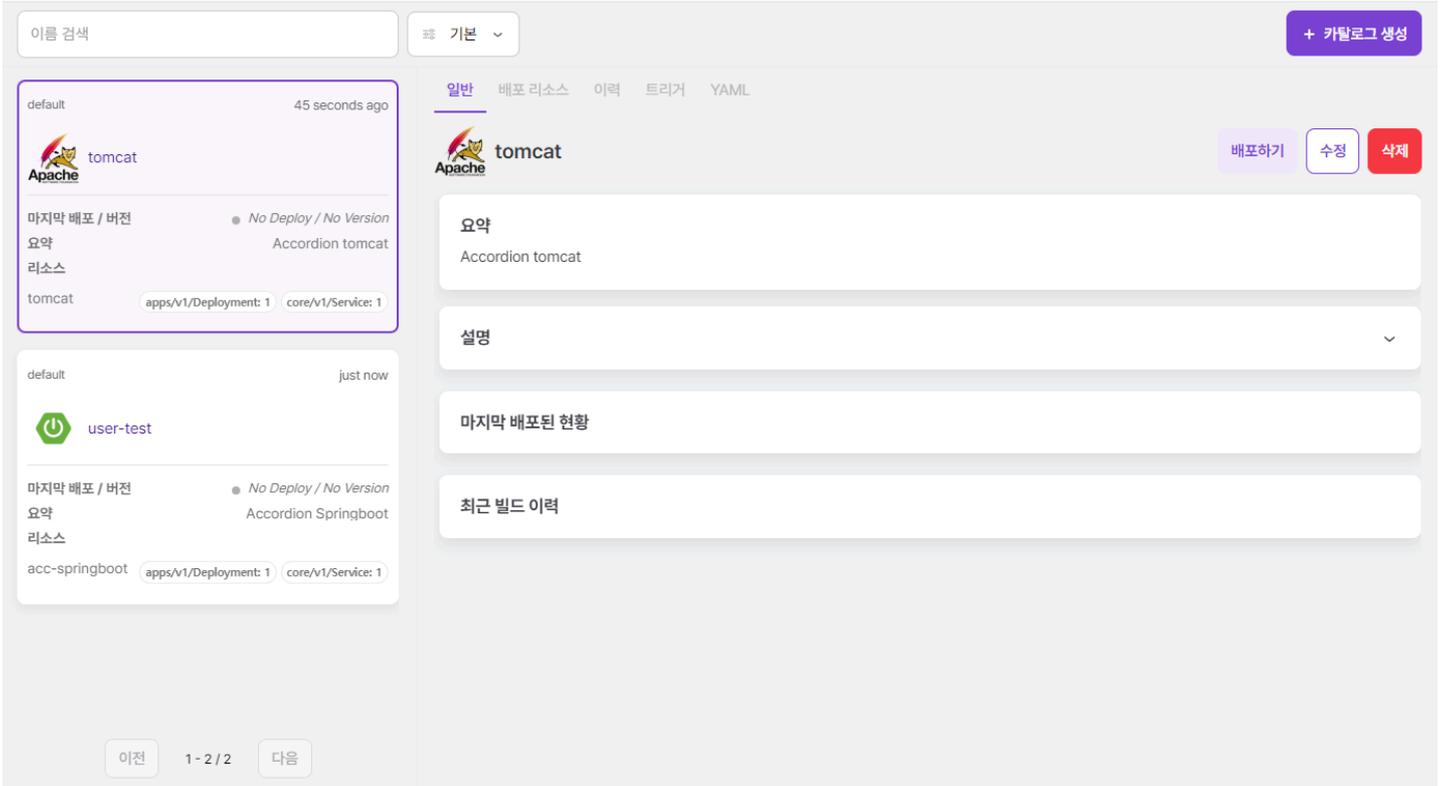
카탈로그는 카탈로그 템플릿을 이용해 쿠버네티스 리소스를 배포하는 아코디언만의 관리 도구이다. 카탈로그의 특징은 다음과 같다.

특징	설명
다양한 쿠버네티스 리소스 배포	디플로이먼트, 스테이트풀셋 등 코어 리소스를 포함하여 커스텀 리소스도 배포할 수 있다.
지속적인 배포	카탈로그로 배포한 리소스 집합에 대해 업그레이드 배포 등 라이프사이클을 관리할 수 있다.
배포 이력 관리	배포를 수행한 시점의 스펙과 수행을 요청한 사용자 등의 이력을 관리한다.
롤백	기존에 성공적으로 배포된 이력에 기술된 배포 리소스 스펙으로 리소스만 다시 생성한다.
다양한 배포 정책	단일 카탈로그를 구성하는 리소스들에 대해서도 리소스 각자의 배포 정책을 가져갈 수 있다.
이력 버전별 비교 및 배포(롤백) 미리보기	배포 리소스 또는 파이프라인의 변경 사항을 버전별로 비교하고, 배포(롤백) 전에 미리보기를 통해 변경 내용을 확인할 수 있다.
멀티클러스터 배포 지원	아코디언에서 관리되는 클러스터에 대해 서로 다른 네임스페이스와 클러스터에 동일한 배포와 라이프사이클을 가져갈 수 있다.
파이프라인 연계	소스에서 컨테이너 이미지를 만들거나, 승인을 포함한 배포 등 다양한 파이프라인 연계를 통해서 배포할 수 있다.

TIP

카탈로그와 카탈로그 템플릿의 관계는 헬름 릴리즈와 헬름 차트의 관계와 유사하다. 카탈로그 템플릿은 배포에 필요한 쿠버네티스 리소스 정보와 변수 정보같은 카탈로그 스펙을 가지고 있다. 사용자가 카탈로그 템플릿으로 카탈로그를 생성하면 시스템이 카탈로그 정보를 이용해 애플리케이션을 배포한다.

애플리케이션 > 카탈로그 > 목록



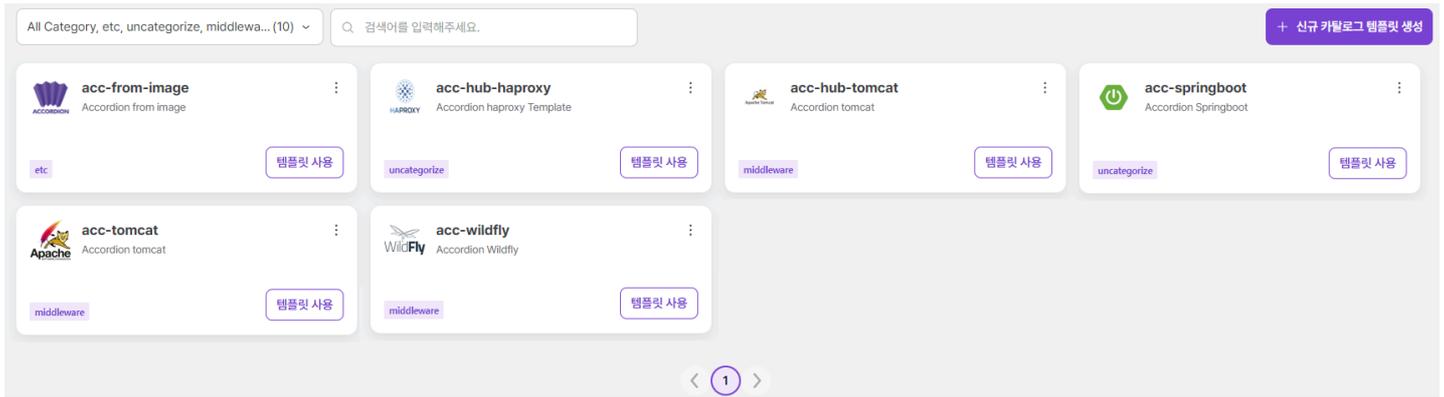
카탈로그 메뉴에서는 카탈로그에 대한 설명과 리소스 구성, 배포 내역을 확인할 수 있다.

항목	설명
일반	카탈로그에 대한 기본 정보와 배포 현황에 대한 정보 제공 <ul style="list-style-type: none"> • 요약 • 설명 • 마지막 배포된 현황 • 최근 빌드 이력
배포 리소스	카탈로그로 배포한 쿠버네티스 리소스 정보 제공
이력	카탈로그로 배포한 이력 정보 제공 및 배포 관리 <ul style="list-style-type: none"> • 카탈로그 배포 과정 조회 • 배포 중인 카탈로그의 배포 중단 • 배포에 성공한 이력에 대해 롤백 • 배포 이력의 변경 사항 버전별 비교
YAML	카탈로그에 대한 쿠버네티스 리소스 관리

항목	설명
트리거	이벤트 발행을 감지하여 카탈로그를 배포 하거나 트리거의 동작을 수행

카탈로그 생성

카탈로그 메뉴의 + 카탈로그 생성 버튼을 클릭하면 카탈로그 생성에 필요한 카탈로그 템플릿을 선택할 수 있다.

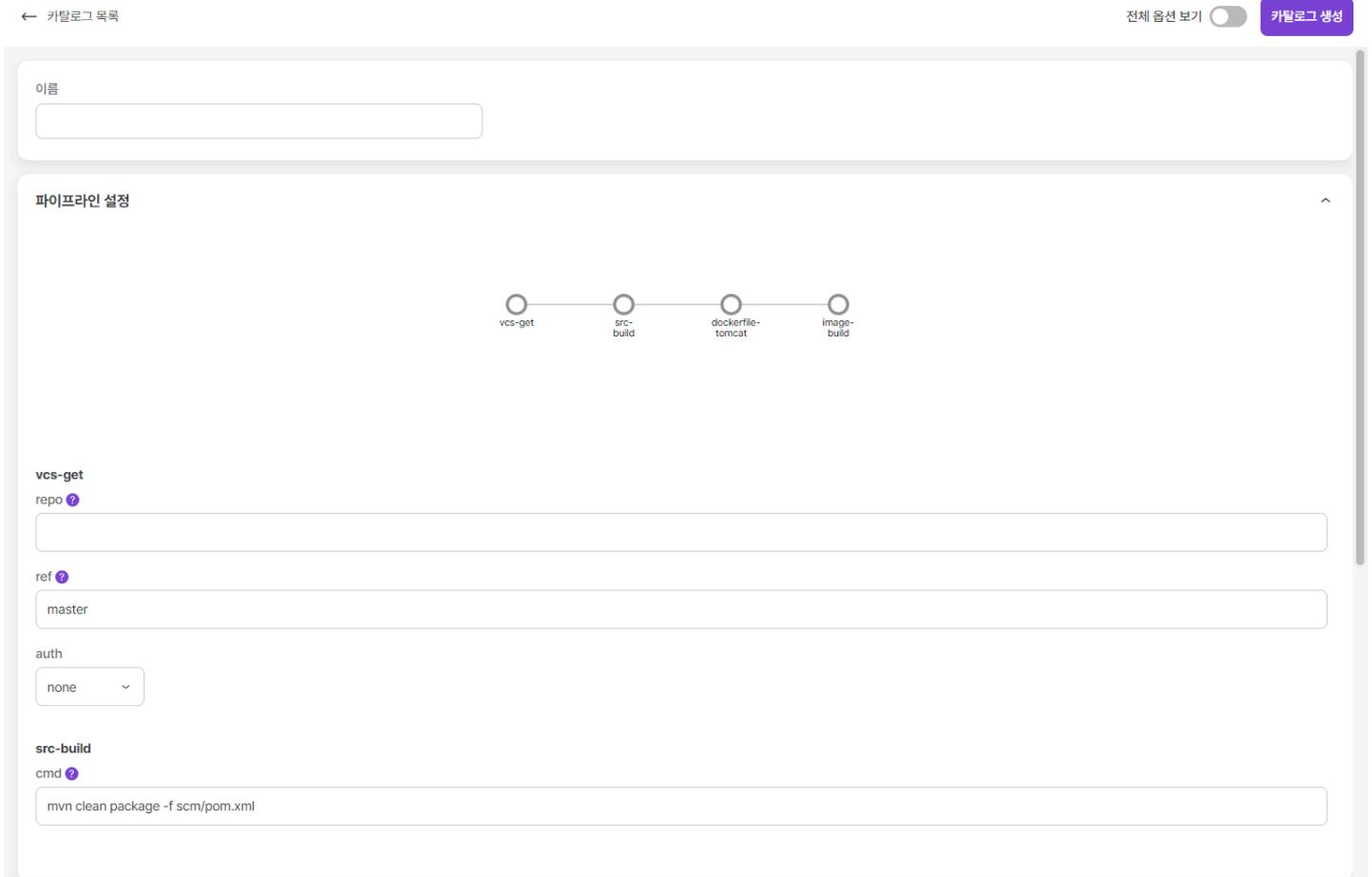


아코디언은 Tomcat, Wildfly 등에 대한 템플릿을 기본 제공하며 카탈로그 템플릿은 추가하거나 변경할 수 있다.

NOTE

만약 템플릿을 사용하지 않고 카탈로그를 생성하고 싶다면 우측 상단의 + 신규 카탈로그 템플릿 생성 버튼을 바로 클릭하여 카탈로그를 직접 생성할 수 있다.

템플릿 사용 버튼을 통해 생성할 카탈로그에 대한 템플릿을 선택하면 카탈로그 설정 화면으로 넘어간다. 다음 화면에서는 카탈로그의 이름과 파이프라인, 배포리소스와 같은 카탈로그를 생성하기 위해 필요한 기본 설정들만 입력한 뒤 우측 상단의 카탈로그 생성 버튼을 클릭하여 카탈로그를 간단하게 생성시킬 수 있다.



우측 상단의 전체 옵션 보기 토글 버튼을 클릭하면 더 자세한 설정을 선택할 수 있다. 해당 토글을 활성화하면 보이는 일반 설정에서는 카탈로그의 이름이나 설명, 로고 이미지와 같은 생성하려는 카탈로그에 대한 기본 정보를 자세히 입력한다.

← 카탈로그 목록 전체 옵션 보기 카탈로그 생성

일반 설정 파이프라인 설정 배포 리소스 설정 YAML

이름

요약

상세 설명 Write Preview

```
acc-tomcat
....
아파치 소프트웨어 재단에서 개발한 서블릿 컨테이너(또는 웹 컨테이너)만 있는 웹 애플리케이션 서버인
아파치 톰캣(Apache Tomcat)을 기반으로 소스빌드(war), 컨테이너 이미지 빌드, 클러스터에 배포 과정(CI/CD)을
포함한 아코디언에서 제공하는 카탈로그입니다.
....
1. 스펙
* J2EE8 Specificataion
* Support for JDK8, 11
* 기본 메모리 설정
* 컨테이너 `1.5 GB` (`resource.limit=1.5Gi`)
* 할당된 컨테이너의 `80%` 를 힙메모리 사용 (`-XX:InitialRAMPercentage=80.0 -XX:MaxRAMPercentage=80.0`)
2. 배포리소스
* Deployemnt
* Service
```

로고 이미지 

항목	설명
이름	카탈로그 이름
요약	카탈로그에 대한 간단한 설명 (생성 후 목록 정보에서 표시)
상세설명	카탈로그에 대한 상세한 설명 (생성 후 마크다운 형식으로 표시)
로고 이미지	카탈로그의 로고 이미지 (미설정할 경우 템플릿의 로고로 설정)

고급 설정 을 선택하면 기본 정보 외 배포 정책 등 상세한 설정이 가능하다. 고급 설정의 항목은 카탈로그마다 조금씩 다를 수 있으며 자주 사용하는 항목은 다음과 같다.

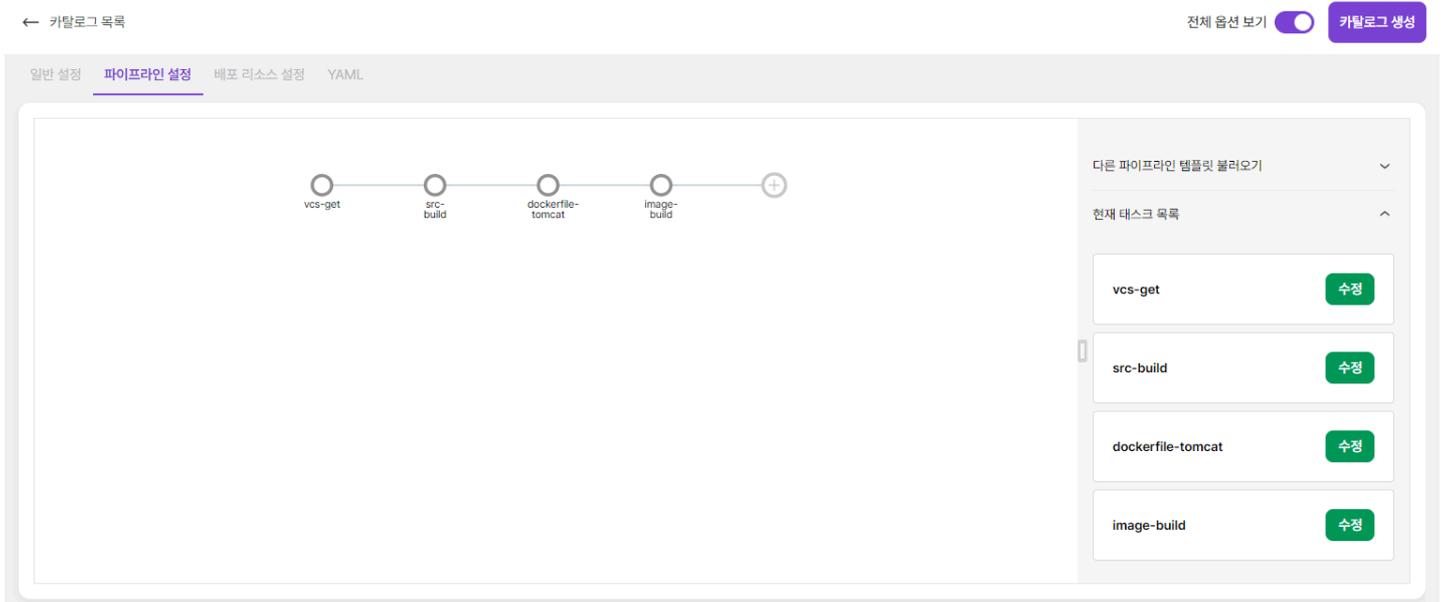
- 배포전략 / 배포 정책 종류

정책	유사한 명령어
리소스가 존재하면 스펙 반영	kubectl apply ...
리소스가 존재하면 스펙 대체	kubectl replace ...
리소스가 존재하면 배포 생략	kubectl create ...
리소스가 존재하면 대체 우선의 업데이트	

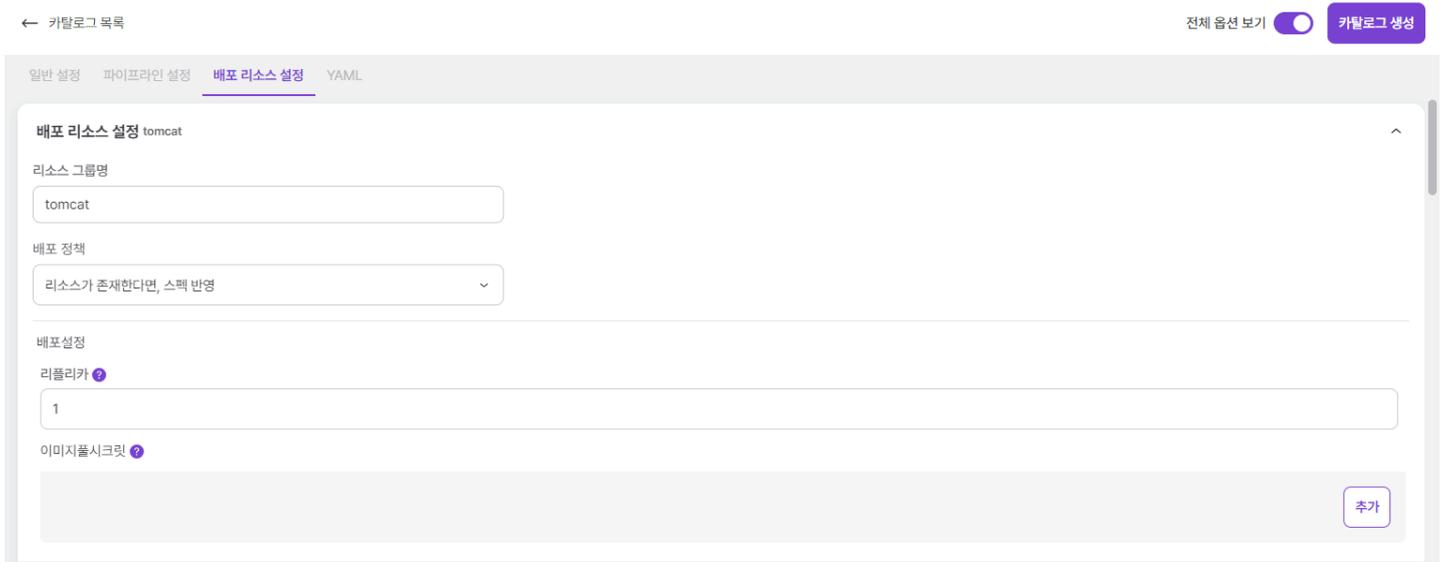
TIP

- 컨테이너 이미지 정책
 - 카탈로그로 애플리케이션을 배포할 때 컨테이너 이미지 빌드를 수행하는 파이프라인을 수행하는 경우 이미지를 저장할 저장소와 최대 보관 개수를 설정한다.
- 배포 이력 보관 정책
 - 카탈로그로 애플리케이션을 배포할 때 배포 이력의 보관 개수와 기간을 설정한다.
 - 자세한 설정은 배포 이력 보관 정책 항목을 참고한다.

상단 탭의 파이프라인 설정 버튼으로 파이프라인에 대한 설정 화면으로 변경할 수 있다. 파이프라인 설정에서는 쿠버네티스 리소스를 배포하기 전에 수행하는 파이프라인에 대해 설정한다. 파이프라인은 기본 파이프라인 템플릿을 가지고 있으며 사용자는 이를 수정하여 배포하는 카탈로그마다 개별 설정할 수 있다.

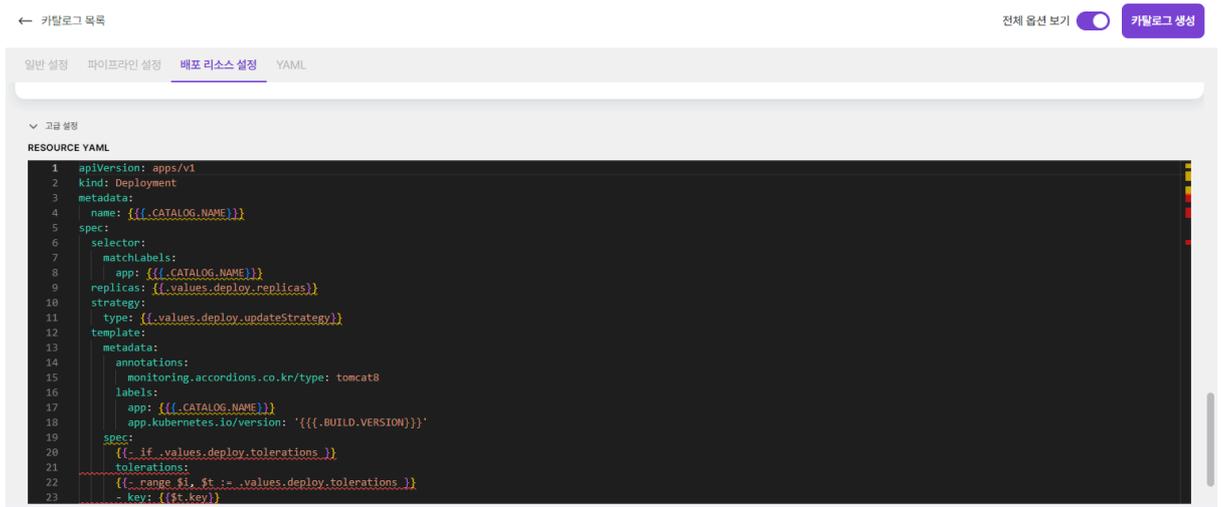


상단 탭 배포 리소스 설정 버튼으로 쿠버네티스에 배포할 리소스에 대한 설정화면으로 변경할 수 있다. 배포 리소스 설정에서는 환경변수 또는 시스템 리소스 할당과 같이 카탈로그로 배포하는 쿠버네티스 리소스에 대한 정보를 입력한다.



고급 설정 을 선택하면 카탈로그로 배포하는 쿠버네티스 리소스에 대해 YAML 편집기에서 설정할 수 있다.

TIP



상단 탭 YAML 버튼으로 카탈로그를 구성하는 정보를 YAML 편집기로 확인하거나 UI에서 설정할 수 없는 정보를 수정할 수 있다.

클러스터 host-cluster-200 네임스페이스 aaaa admin

← 카탈로그 목록 전체 옵션 보기 카탈로그 생성

일반 설정 파이프라인 설정 배포 리소스 설정 **YAML**

```

1  apiVersion: ckd.accordions.co.kr/v1beta1
2  kind: Catalog
3  metadata:
4    annotations:
5      accordions.co.kr/description: |
6        acc-tomcat
7        ----
8        ...
9        이파치 소프트웨어 재단에서 개발한 서블릿 컨테이너(또는 웹 컨테이너)만 있는 웹 애플리케이션 서버인
10       이파치 톰캣(Apache Tomcat)을 기반으로 소스빌드(war), 컨테이너 이미지 빌드, 클러스터에 배포 과정(CI/CD)을
11       포함한 아코디언에서 제공하는 카탈로그입니다.
12       ...
13       1. 스펙
14         * J2EE8 Specificataion
15         * Support for JDK8, 11
16         * 기본 메모리 설정
17         * 컨테이너 '1.5 GB' ('resource.limit=1.5Gi')
18         * 할당된 컨테이너의 '80%' 를 힙메모리 사용 ('-XX:InitialRAMPercentage=80.0 -XX:MaxRAMPercentage=80.0')
19       2. 배포리소스
20         * Deployemnt
21         * Service
22       accordions.co.kr/logo.png-data: >-
23         iVBORw0KGgoAAAANSUHEUgAAAGQAAABkCATAAAD/gAIDAAAw/k1EQVR4nOx9B5hV1bX/3vuU28vce6f33mAmZAiZWBAsRcS0YiKDRMj8pJoxP6+ZyXRqDHJ89mSa0yCIEVAyjD80gxMgYHpvD/ey2l7/7977zACgo8kQmz7/1v8vrk
24       accordions.co.kr/logo.resized: 'true'
25       accordions.co.kr/logo.size: 100,100
26       accordions.co.kr/summary: Accordion tomcat
27       ckd.accordions.co.kr/retention.count: '50'
28       ui.accordions.co.kr/category: middleware
29     labels: {}
30     name: ''
31     namespace: aaaa
32   spec:
33     deployStrategy:
34       defaultPolicy: Apply
35       image:
36         archiveCount: 5
37         registryName: user-registry
38       clusters:
39         - name: localcluster
40           namespaces:
41             - aaaa
42     pipelineTemplateRef:
43       name: acc-vcs-jbuild-tomcat
44     resourceValues:
45       - name: tomcat

```

> 이벤트 (NORMAL 54 / WARNING 42)

YAML 편집기를 통해 카탈로그를 보면 예약된 환경변수를 볼 수 있다. 카탈로그에서 예약된 환경변수는 다음과 같다.

이름	환경변수 표현식	추가설명
카탈로그 이름	{{{.CATALOG.NAME}}}	카탈로그의 이름
카탈로그 네임스페이스	{{{.CATALOG.NAMESPACE}}}	카탈로그가 배포된 네임스페이스 이름
이미지 레지스트리	{{{.IMAGE.REGISTRYNAME}}}	<ul style="list-style-type: none"> catalog.spec.deployStrategy.image.registryName 카탈로그를 사용하지 않는 경우 공백의 기본값
이미지 이름	{{{.IMAGE.NAME}}}	<ul style="list-style-type: none"> {{{.IMAGE.REGISTRYNAME}}} 가 존재하는 경우 설정 이름 생성 규칙은 하단 참고 카탈로그를 사용하지 않는 경우 공백의 기본값
이미지 태그	{{{.IMAGE.TAG}}}	<ul style="list-style-type: none"> 명시적으로 입력하면, {{{.IMAGE.NAME}}} 에서 태그 고정
이미지 이름 패턴	{{{.IMAGE.NAMEPATTERN}}}	<ul style="list-style-type: none"> gen-full (기본값) <ul style="list-style-type: none"> {{{.IMAGE.NAME}}} 전체 자동 설정 태그는 {{{.BUILD.VERSION}}} 으로 설정되며, 다시빌드(recreate)시, 태그를 현재 버전으로 변경 이 경우에만 보관개수 (catalog.spec.deployStrategy.image.archiveCount)에 따른 삭제 수행 gen-name <ul style="list-style-type: none"> {{{.IMAGE.TAG}}} 가 존재하여 {{{.IMAGE.NAME}}} 태그를 제외한 이름 자동 설정 custom <ul style="list-style-type: none"> 명시적으로 {{{.IMAGE.NAME}}} 기술
레지스트리 서버	{{{.REGISTRY.SERVER}}}	{{{.IMAGE.REGISTRYNAME}}} secret의 server
레지스트리 사용자이름	{{{.REGISTRY.USERNAME}}}	{{{.IMAGE.REGISTRYNAME}}} secret의 username

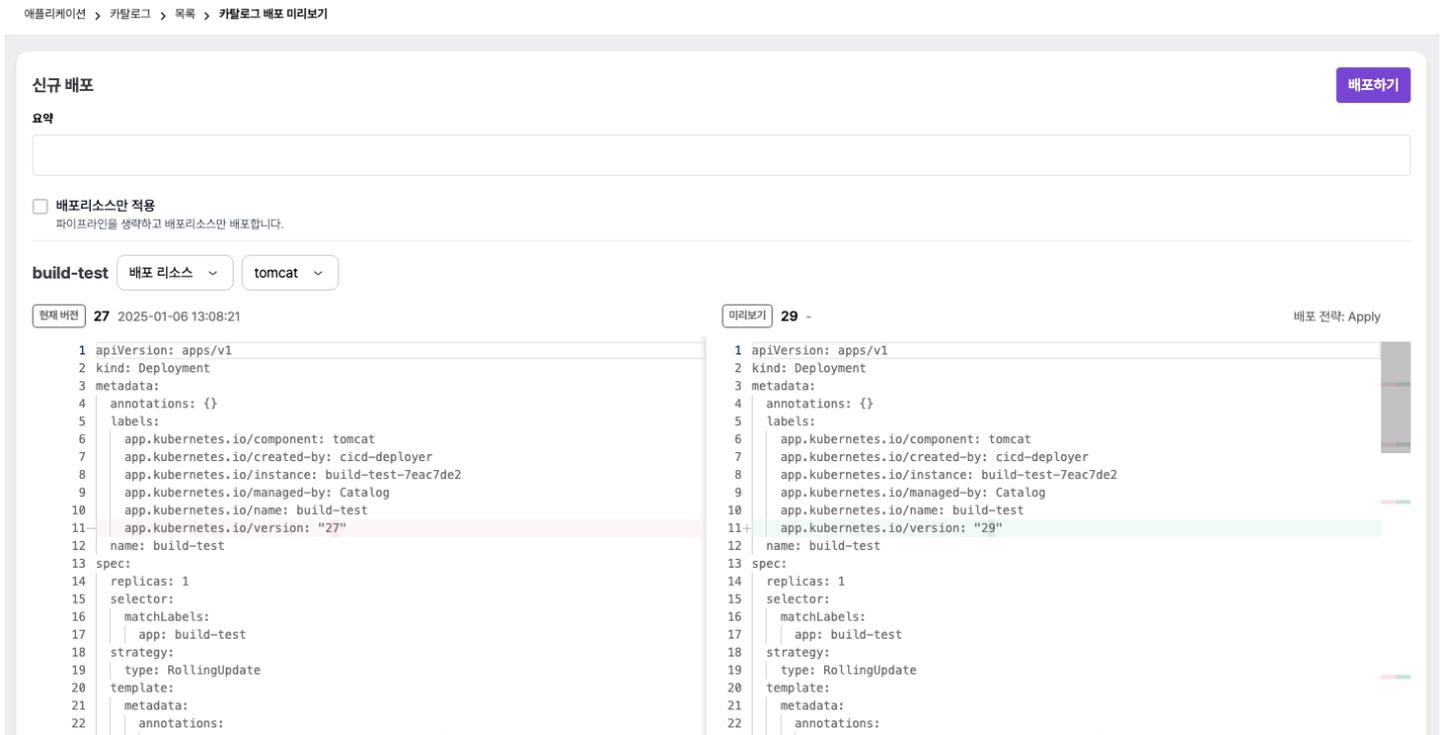
이미지 이름 환경변수는 레지스트리와 사용자 환경변수에 따라 적용된다.

이미지 이름의 생성 규칙과 관련 환경변수 설정은 다음과 같다.

INPUT	OUTPUT
<p>일반 형식</p> <ul style="list-style-type: none"> (registry) server : 127.0.0.1:5000 (registry) username : acc {{{.PIPELINE.INSTANCE}}} : tmapp-c834f245 {{{.BUILD.VERSION}}} : 2 	<ul style="list-style-type: none"> {{{.REGISTRY.SERVER}}} : 127.0.0.1:5000 {{{.REGISTRY.USERNAME}}} : acc {{{.IMAGE.NAMEPATTERN}}} : gen-full {{{.IMAGE.NAME}}} : 127.0.0.1:5000/acc/tmapp-c834f245:2
<p>Dockerhub</p> <ul style="list-style-type: none"> (registry) server : https://index.docker.io/v1/ (registry) username : acc {{{.PIPELINE.INSTANCE}}} : tmapp-c834f245 {{{.BUILD.VERSION}}} : 2 	<ul style="list-style-type: none"> {{{.REGISTRY.SERVER}}} : docker.io {{{.REGISTRY.USERNAME}}} : acc {{{.IMAGE.NAMEPATTERN}}} : gen-full {{{.IMAGE.NAME}}} : docker.io/acc/tmapp-c834f245:2
<p>스코프를 포함하는 경우</p> <ul style="list-style-type: none"> (registry) server : registry.gitlab.com/scope/test (registry) username : acc {{{.PIPELINE.INSTANCE}}} : tmapp-c834f245 {{{.BUILD.VERSION}}} : 2 	<ul style="list-style-type: none"> {{{.REGISTRY.SERVER}}} : registry.gitlab.com/scope/test {{{.REGISTRY.USERNAME}}} : acc {{{.IMAGE.NAMEPATTERN}}} : gen-full {{{.IMAGE.NAME}}} : registry.gitlab.com/scope/test/tmapp-c834f245:2
<p>이미지 태그를 사용하는 경우</p> <ul style="list-style-type: none"> (registry) server : 127.0.0.1:5000 (registry) username : acc {{{.PIPELINE.INSTANCE}}} : tmapp-c834f245 {{{.BUILD.VERSION}}} : 2 {{{.IMAGE.TAG}}} : dev 	<p>이미지 이름의 태그 변경 : 빌드버전 ⇒ 태그</p> <ul style="list-style-type: none"> {{{.REGISTRY.SERVER}}} : 127.0.0.1:5000 {{{.REGISTRY.USERNAME}}} : acc {{{.IMAGE.TAG}}} : dev {{{.IMAGE.NAMEPATTERN}}} : gen-name {{{.IMAGE.NAME}}} : 127.0.0.1:5000/acc/tmapp-c834f245:dev
<p>이미지 이름을 사용하는 경우</p> <ul style="list-style-type: none"> (registry) server : 127.0.0.1:5000 (registry) username : acc {{{.PIPELINE.INSTANCE}}} : tmapp-c834f245 {{{.IMAGE.NAME}}} : 127.0.0.1:5000/change/custom-app:latest 	<p>설정된 이미지 이름 고정</p> <ul style="list-style-type: none"> {{{.REGISTRY.SERVER}}} : 127.0.0.1:5000 {{{.REGISTRY.USERNAME}}} : acc {{{.IMAGE.NAMEPATTERN}}} : custom {{{.IMAGE.NAME}}} : 127.0.0.1:5000/change/custom-app:latest

INPUT	OUTPUT
<p>이미지 이름과 이미지 태그를 같이 사용하는 경우</p> <ul style="list-style-type: none">• (registry) server : 127.0.0.1:5000• (registry) username : acc• {{{.PIPELINE.INSTANCE}}} : tmapp-c834f245• {{{.IMAGE.TAG}}} : dev• {{{.IMAGE.NAME}}} : 127.0.0.1:5000/change/custom-app:latest	<p>이미지 태그 무시</p> <ul style="list-style-type: none">• {{{.REGISTRY.SERVER}}} : 127.0.0.1:5000• {{{.REGISTRY.USERNAME}}} : acc• {{{.IMAGE.NAMEPATTERN}}} : custom• {{{.IMAGE.NAME}}} : 127.0.0.1:5000/change/custom-app:latest

모든 설정이 끝나면 카탈로그 생성 버튼을 선택하여 카탈로그 생성을 마친다. 카탈로그가 생성되었다고하여 실제 애플리케이션이 배포된 것은 아니다. 카탈로그 목록에서 방금 작성한 카탈로그를 선택하고 배포하기 버튼을 선택하면 신규 배포 전 변경 사항을 현재 버전과 미리보기 버전을 비교할 수 있는 Diff UI 화면으로 이동한다.

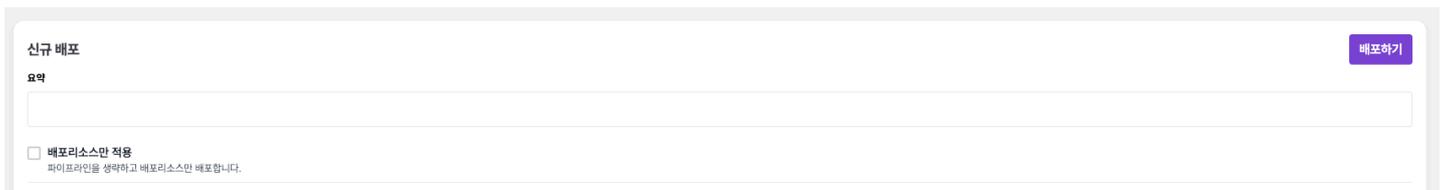


실제 배포가 수행되면 적용될 변경 사항을 현재 버전과 비교하여 미리보기 기능을 제공한다. 배포 리소스, 파이프라인 각각의 변경 사항을 확인할 수 있으며 배포리소스일 경우 우측의 리소스 그룹 을 선택하여 리소스 그룹별로 변경 사항을 확인할 수 있다.

해당 페이지에서 우측 상단의 배포하기 버튼을 선택해야 파이프라인을 통해 애플리케이션이 배포된다.

배포리소스만 적용 을 선택하여 배포할 경우 가장 최근 배포한 빌드의 이미지를 사용하여 현재 카탈로그에 정의된 스펙으로 리소스 배포만 진행 한다. 가장 최근 배포한 빌드의 이미지는 2.9.0 업그레이드 이후 배포한 이미지만 해당한다.

2.9.0 이전 버전의 빌드 이미지로는 배포리소스만 적용 하여 배포가 불가능하며, 해당 기능을 사용하려면 업그레이드 이후 최소 1번의 배포를 진행해야 한다.

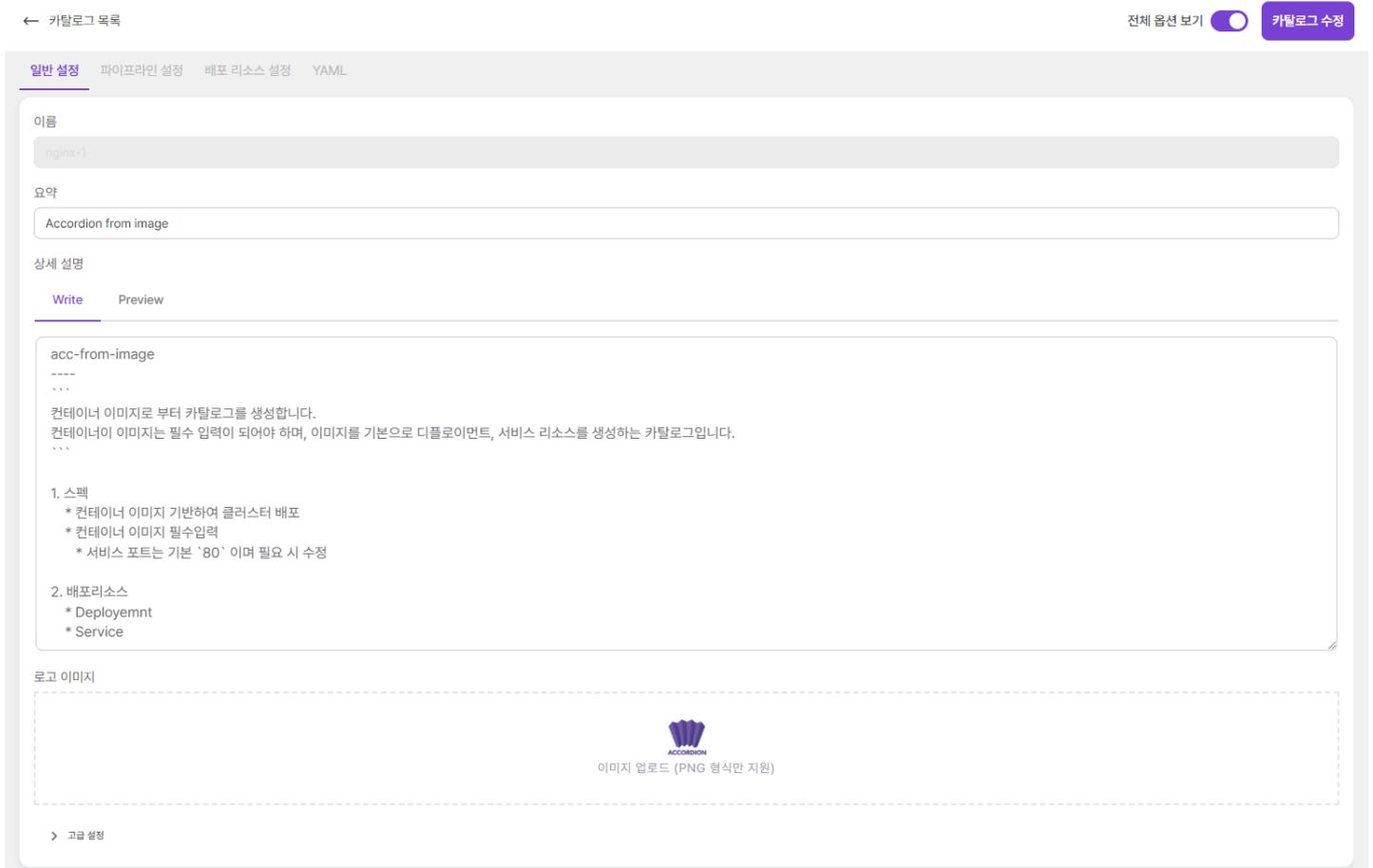


TIP

배포시 요약 필드는 필수값은 아니지만 애플리케이션 관리를 위해 설정을 권장한다.

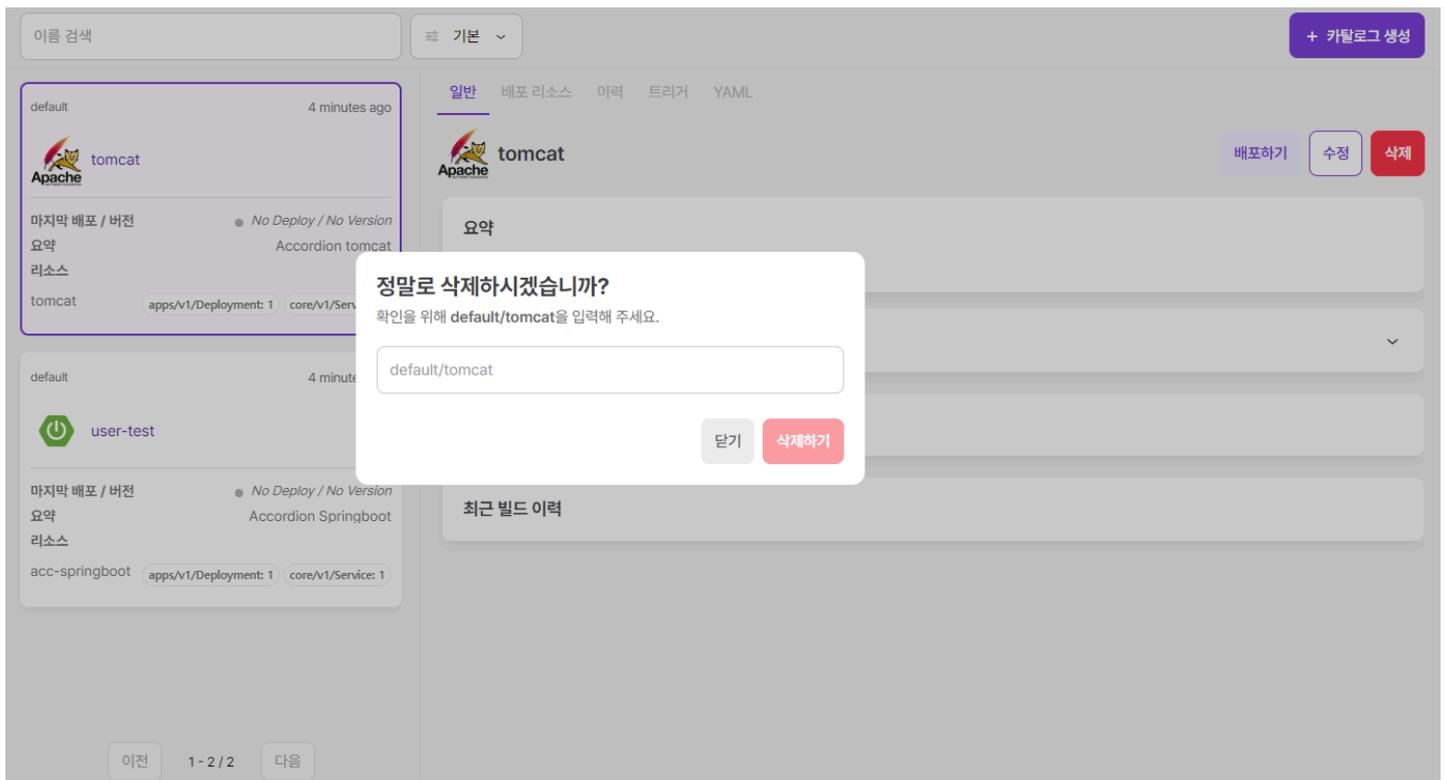
카탈로그 수정

카탈로그를 변경하기 위해서 목록에서 카탈로그를 찾아 수정 버튼을 선택한다. 카탈로그 수정 시 설정값은 앞에 카탈로그 생성 시 입력했던 값과 유사하다.



카탈로그 삭제

삭제하려는 카탈로그를 선택하고 우측의 삭제 버튼을 선택한다.



모달에서 네임스페이스와 카탈로그 이름을 입력하여 삭제한다.

이력

배포한 카탈로그의 이력들을 확인할 수 있다.

특정 이력에 대한 롤백 기능, 이력별 변경 사항 확인 및 해당 이력들을 csv 파일로 추출하는 내보내기 기능을 제공한다.

롤백의 경우, 배포리소스만 적용 기능을 사용할 수 없다.

The screenshot shows the 'build-test' deployment history page. On the left, there is a summary card for the 'tomcat' resource, indicating it is 'Terminated / 28' and listing 'apps/v1/Deployment: 1' and 'core/v1/Service: 1'. The main area displays a table of deployment records:

상태	버전 ↓	실행자	시작 시간	종료 시간	메시지	동작
○	28	accordion:user:admin	2025-01-06 13:13:37	2025-01-06 13:13:37		롤백 변경사항
●	27	accordion:user:admin	2025-01-06 13:08:21	2025-01-06 13:10:25		롤백 변경사항
○	26	accordion:user:admin	2025-01-06 13:08:10	2025-01-06 13:08:10		롤백 변경사항
●	25	accordion:user:admin	2025-01-06 11:26:57	2025-01-06 11:28:52		롤백 변경사항
○	24	accordion:user:admin	2025-01-06 11:22:48	2025-01-06 11:22:48		롤백 변경사항
○	23	accordion:user:admin	2025-01-06 11:13:00	2025-01-06 11:13:00		롤백 변경사항
○	22	accordion:user:admin	2025-01-06 11:12:48	2025-01-06 11:12:48		롤백 변경사항
○	21	accordion:user:admin	2025-01-06 11:11:54	2025-01-06 11:11:54		롤백 변경사항
○	20	accordion:user:admin	2025-01-06 11:11:40	2025-01-06 11:11:40		롤백 변경사항

이력별 변경 사항 확인

이력에서 선택한 특정 버전을 기준으로 성공한 다른 버전과의 변경 사항을 비교할 수 있다.

빌드가 성공한 버전에 한하여 선택이 가능하며 변경사항 버튼을 클릭하여 비교 페이지로 이동할 수 있다.

The screenshot shows the '이력 변경사항' (History Comparison) page. It compares two deployment manifests side-by-side. The left manifest is for version 25 (created at 11:26:56) and the right is for version 19 (created at 11:09:04). The comparison highlights differences in the 'app.kubernetes.io/version' field, which is '25' in the left manifest and '19' in the right manifest. The rest of the manifest structure, including labels, spec, and containers, is identical.

비교는 배포 리소스와 파이프라인 두 가지 타입에 한해 가능하며, 배포 리소스의 경우 하위 리소스 그룹별로 변경 사항을 비교할 수 있다.

기본값은 배포 리소스 비교이며, 비교 기준이 되는 버전의 policy가 Redeploy 또는 Deploy가 아닌 경우에만 파이프라인 비교가 가능하다.

또한, 선택 가능한 비교 버전은 빌드가 성공한 버전에 한하며, 파이프라인의 경우 추가적으로 policy가 Redeploy 또는 Deploy가 아닌 버전만 비교할 수 있다.

롤백 미리보기

선택한 특정 버전으로 롤백을 수행하기 전에 변경 사항을 미리 확인할 수 있다.
롤백 버튼을 클릭하면 롤백 전 변경 사항을 확인할 수 있는 미리보기 페이지로 이동한다.

클러스터 jhcae-host-cluster 네임스페이스 default admin

애플리케이션 > 카탈로그 > 목록 > 카탈로그 롤백 미리보기

build-test 배포 리소스 tomcat 롤백하기

현재 버전 27 2025-01-06 13:08:21 배포 전략: Apply 롤백할 버전 19 2025-01-06 11:09:04 배포 전략: Apply

```

1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4 annotations: {}
5 labels:
6 app.kubernetes.io/component: tomcat
7 app.kubernetes.io/created-by: cicd-deployer
8 app.kubernetes.io/instance: build-test-7eac7de2
9 app.kubernetes.io/managed-by: Catalog
10 app.kubernetes.io/name: build-test
11 app.kubernetes.io/version: "27"
12 name: build-test
13 spec:
14 replicas: 1
15 selector:
16 matchLabels:
17 app: build-test
18 strategy:
19 type: RollingUpdate
20 template:
21 metadata:
22 annotations:
23 monitoring.accordions.co.kr/type: tomcat8
24 labels:
25 app: build-test
26 app.kubernetes.io/version: "27"
27 spec:
28 containers:
29 - env:
30 - name: DEFAULT_HEAP_OPTS
31 value: ""
32 - name: JAVA_HEAP_OPTS
33 value: -XX:InitialRAMPercentage=80.0 -XX:MaxRAMPercentage=80.0
34 - name: JAVA_OPTS
35 value: -Duser.timezone=GMT+09:00 $(JAVA_HEAP_OPTS) $(DEFAULT_HEAP_OPTS)
36 -XX:+UseParallelGC -XX:+PrintGCDetails -XX:+HeapDumpOnOutOfMemoryError

```

```

1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4 annotations: {}
5 labels:
6 app.kubernetes.io/component: tomcat
7 app.kubernetes.io/created-by: cicd-deployer
8 app.kubernetes.io/instance: build-test-7eac7de2
9 app.kubernetes.io/managed-by: Catalog
10 app.kubernetes.io/name: build-test
11 app.kubernetes.io/version: "19"
12 name: build-test
13 spec:
14 replicas: 1
15 selector:
16 matchLabels:
17 app: build-test
18 strategy:
19 type: RollingUpdate
20 template:
21 metadata:
22 annotations:
23 monitoring.accordions.co.kr/type: tomcat8
24 labels:
25 app: build-test
26 app.kubernetes.io/version: "19"
27 spec:
28 containers:
29 - env:
30 - name: DEFAULT_HEAP_OPTS
31 value: ""
32 - name: JAVA_HEAP_OPTS
33 value: -XX:InitialRAMPercentage=80.0 -XX:MaxRAMPercentage=80.0
34 - name: JAVA_OPTS
35 value: -Duser.timezone=GMT+09:00 $(JAVA_HEAP_OPTS) $(DEFAULT_HEAP_OPTS)
36 -XX:+UseParallelGC -XX:+PrintGCDetails -XX:+HeapDumpOnOutOfMemoryError

```

가장 최근에 빌드가 성공한 버전이 현재 버전 으로 설정되며, 이 버전이 비교 기준이 된다. 이력에서 선택한 특정 버전은 롤백할 버전 으로 지정되어 비교가 이루어진다.

카탈로그 배포 이력을 선택하면 해당 카탈로그 이력에 대한 단계별 세부 상태와 로그를 확인할 수 있다.

애플리케이션 > 카탈로그 > 목록

build-test 기본 + 카탈로그 생성

일반 배포 리소스 이력 트리거 YAML

build-test 배포하기 수정 삭제

마지막 배포 / 버전 Terminated / 28
요약 리소스 Accordion tomcat

tomcat apps/v1/Deployment:1 core/v1/Service:1

이전 1-1/1 다음

뒤로가기

상태	버전 ↓	실행자	시작 시간	종료 시간	메시지	종류
✓	27	accordion:user:admin	2025-01-06 13:08:21	2025-01-06 13:10:25		롤백 변경사항

파이프라인 YAML

```

graph LR
  test[✓ test] --> vcs-get[✓ vcs-get]
  vcs-get --> src-build[✓ src-build]
  src-build --> dockerfile-tomcat[✓ dockerfile-tomcat]
  dockerfile-tomcat --> image-build[✓ image-build]
  image-build --> deploy-localcluster-default[✓ deploy-localcluster-default]

```

- ✓ > TEST 4 seconds
- ✓ > VCS-GET 8 seconds
- ✓ > SRC-BUILD 21 seconds
- ✓ > DOCKERFILE-TOMCAT 4 seconds
- ✓ > IMAGE-BUILD 15 seconds

트리거

애플리케이션 > 카탈로그 > 목록

The screenshot shows the 'tomcat' trigger configuration page. It includes a search bar with 'tomcat', a '기본' (Basic) dropdown, and a '+ 카탈로그 생성' (Create Catalog) button. The main content area has tabs for '일반' (General), '배포 리소스' (Deployment Resources), '이력' (History), '트리거' (Triggers), and 'YAML'. Under the '트리거' tab, there are two trigger configurations:

- 예약 액션 (Scheduled Action):** Name 'test', status '실행' (Running), '변경 상태' (Change State) '상태 없음' (None), '최근 트리거' (Recent Triggers) '없음' (None), '수행 주기' (Execution Cycle) '0 0 1 * *', with '수정' (Edit) and '삭제' (Delete) buttons.
- 빌드 후 액션 (Post-Build Action):** Name 'test-test', status '실행' (Running), '변경 상태' (Change State) '상태 없음' (None), '최근 트리거' (Recent Triggers) '없음' (None), with '수정' (Edit) and '삭제' (Delete) buttons.

At the bottom, there are navigation buttons: '이전' (Previous), '1 - 1 / 1', and '다음' (Next).

트리거는 카탈로그/파이프라인을 대상으로 하며, 이벤트 발생을 기준으로 2가지로 나뉜다.

종류	설명
예약 액션 트리거	주기적으로 등록된 이벤트를 감지, 대상이 되는 카탈로그/파이프라인을 실행
빌드 후 액션 트리거	카탈로그/파이프라인이 배포 또는 빌드된 이후 동작을 수행

예약 액션 트리거

예약 액션 트리거는 등록된 스케줄에 따라 주기적으로 동작하며 **이벤트가 감지** 되면 빌드(배포)를 수행한다.

아래는 사용 가능한 예약 액션 트리거의 종류이다.

트리거 이름	설명
버전관리 시스템의 저장소 변경 감지 트리거	빌드의 메타데이터를 기반으로 원격저장소의 변경을 감지
컨테이너 이미지 변경 감지 트리거	빌드의 메타데이터를 기반으로 컨테이너 이미지 변경을 감지 또는 빌드의 <code>spec.env.EXTERNAL IMAGE</code> 정보를 기반으로 이미지의 변경을 감지
http 요청에 대한 응답 감지 트리거	http 요청을 통해서 응답 데이터를 감지
정기적으로 트리거 실행	등록된 스케줄 주기 마다 빌드(배포)를 수행

스케줄

- [Cron](https://ecosia.org) (<https://ecosia.org>) 방식으로 등록한다.

- 스케줄의 경우, **3-5분 이상** 으로 등록하길 권장한다.

- 기본값 : 0/5 * * * *

버전관리 시스템의 저장소 변경 감지 트리거

vcs 태스크의 메타데이터를 원격 저장소와 비교하여, 변경 사항을 감지한다.

git 또는 svn 을 지원한다.

해당 트리거는 빌드 메타데이터를 기반으로 동작하며, vcs에 따라 아래와 같은 필수값이 필요하다.

git 는 아래와 같다.

메타데이터	설명	예시
vcs	git	git
repo	원격 저장소	https://github.com/mantech-accordion/sample-war.git
ref	저장소 레퍼런스	master
commit	커밋 정보	320a4819fe5b5dcb1892e263620d79849b83012e

svn 는 아래와 같다.

메타데이터	설명	예시
vcs	svn	svn
repo	원격 저장소	svn://acc@xx.xx.xx.xx/hello
revision	리비전	5

컨테이너 이미지 변경 감지 트리거

컨테이너 이미지의 변경 사항을 감지한다.

컨테이너 내부/외부 이미지 설정을 통해 감지할 이미지 대상을 설정할 수 있다.

내부 이미지 감지

내부 이미지 설정 의 지원되는 레지스트리 저장소는 DockerHub 또는 Registry API 2.0 을 지원해야 한다.

내부 이미지 설정 의 메타데이터 필수값은 아래와 같다.

메타데이터	예시	설명
image	docker.io/accordions/tmapp-03404282:2	컨테이너 이미지
digest	sha256:13be6694216a8bdc62fcb6c112b2165ff43341eddbddb88a39a17329d5bb8aaa	이미지 digest로 mode 가 digest 인 경우 필수

외부 이미지 감지

catalog 를 대상으로만 동작한다.

외부 이미지 감지 는 내부 이미지 감지 와 달리 빌드의 메타데이터에서 정보를 가져오는 것이 아닌 아래와 같이 빌드의

spec.env.EXTERNAL_IMAGE 정보를 통해 이미지 변경을 감지한다.

```
spec:
  env:
    EXTERNAL IMAGE:
      DIGEST: sha256:9822b71cd6b28545703fde132adb98cd1057ea387e41c1d84a254486feb3c2cc
      NAME: image
      REGISTRYNAME: user-registry
```

catalog 의 spec.deployStrategy.externalImage 에 아래와 같이 등록할 정보를 입력하고 빌드 또는 배포하면 위와 같이 빌드의 spec.env.EXTERNAL IMAGE 에 정보가 등록된다.

```
spec:
  deployStrategy:
    externalImage:
      name: "image"
      registryName: user-registry
```

http 요청에 대한 응답 감지 트리거
외부 http 요청 이후 응답데이터를 비교하여 같은 경우 빌드(배포)를 수행

응답 콘텐츠는 json 형식만 지원 : Content-Type: application/json

응답 상태 코드는 200 만 지원

정기적으로 트리거 실행 트리거
설정된 스케줄에 따라 주기적으로 빌드(배포)를 수행
빌드 후 액션 트리거
사용 가능한 트리거의 종류이다.

트리거 이름	설명
카탈로그/파이프라인 빌드(배포) 트리거	같은 클러스터의 다른 카탈로그/파이프라인을 빌드(배포)
웹훅 전달 트리거	대상 객체와 빌드 정보를 http로 보내준다.

카탈로그/파이프라인 빌드(배포) 트리거
빌드의 상태를 비교하여 같은 클러스터의 다른 카탈로그/파이프라인을 빌드(배포) 한다.

웹훅 전달 트리거
보내주는 데이터 형식은 아래와 같다.

```
{
  "object": {카탈로그 or 파이프라인},
  "build": {마지막 빌드}
}
```

트리거 등록

이름 검색 기본 + 카탈로그 생성

alert 2 months ago

jpetstore-6

마지막 배포 / 버전 ● Succeeded / 1

요약 Accordion tomcat

리소스

tomcat apps/v1/Deployment: 1 core/v1/Service: 1

alert 7 months ago

tomcat-jpetstore

마지막 배포 / 버전 ● Failed / 5

요약 Accordion tomcat

리소스

tomcat

일반 배포 리소스 이력 트리거 YAML

jpetstore-6 트리거 생성

등록된 트리거가 없어요. 트리거를 추가해 주세요.

이전 1 - 2 / 2 다음

트리거 탭을 선택 한 뒤 트리거 생성 버튼을 클릭

예약 액션 트리거 또는 빌드 후 액션 트리거 를 선택한다.

예약 액션 빌드 후 액션

이름

트리거 이름을 입력해 주세요.

스케줄

스케줄을 입력해 주세요. ▼

버전관리 시스템 저장소 변경 감지

컨테이너 이미지 변경 감지

http 요청에 대한 응답 감지

정기적으로 트리거 실행

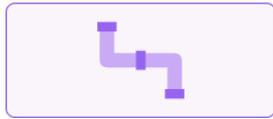
트리거 수행 서비스 어카운트 ▼

[+ 옵션 추가하기](#)

예약 액션 빌드 후 액션

이름

트리거 이름을 입력해 주세요.



카탈로그/파이프라인 빌드 트리거



웹훅 전달 트리거

트리거 수행 서비스 어카운트

▼

대상(카탈로그/파이프라인) 종류

▼

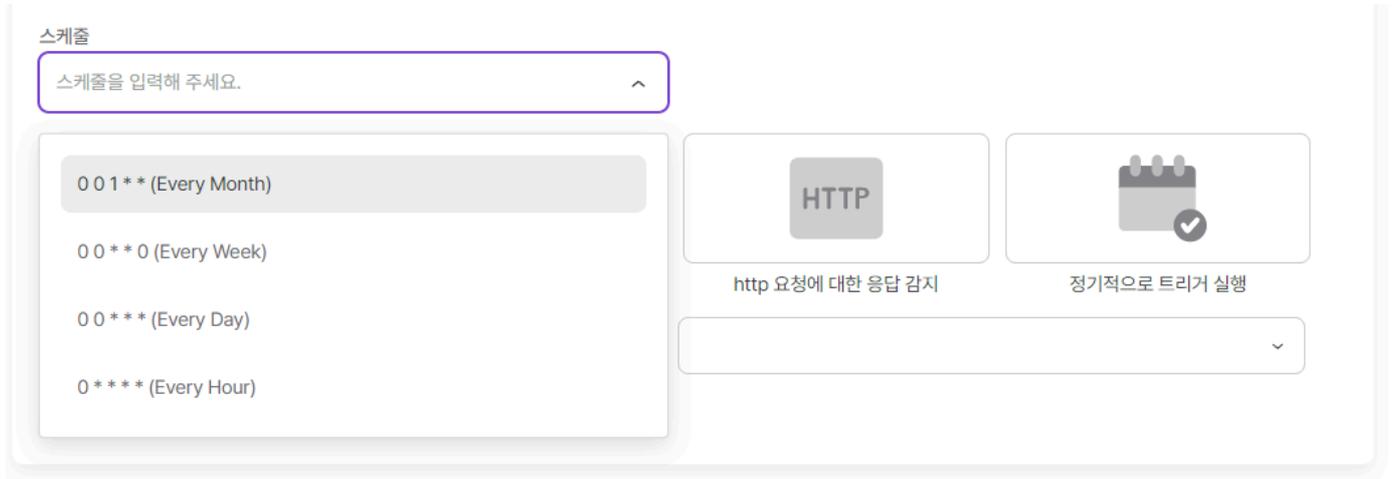
빌드(배포)를 수행할 대상 지정

대상(카탈로그/파이프라인) 종류 설정이 필요합니다. ▼

+ 옵션 추가하기

클릭하면 이름, 타입, 옵션 을 입력할 수 있는 화면이 나타난다.

- 이름은 해당 트리거에 대한 이름을 설정할 수 있다
- 타입은 트리거의 타입을 선택할 수 있다.
- 옵션은 선택한 트리거에서 필요한 옵션의 키와 밸류를 입력 한다.



예약 액션 트리거 등록시에는 스케줄 선택박스를 클릭하면 기본으로 설정되어 있는 스케줄을 설정할 수 있다.

직접 스케줄 설정 작성시 아래와 같이 크론 스케줄 포맷에 의해 작성할수도 있다.

시간	허용 가능 값	허용 가능 특수문자
분	0-59	*, /, -
시	0-23	*, /, -
일자	1-13	*, /, - ?
달	1-12 or JAN-DEC	*, /, -
요일	0-6 or SUN-SAT	*, /, - ?

스케줄 설정의 예는 아래와 같다.

1 9 1 11 *

= 매년 11월 1일 오전 9시 1분에 동작

원하는 트리거 선택 및 키 밸류 값을 입력 후 우측 상단의 트리거 생성 버튼을 클릭한다.

버전관리 시스템의 저장소 변경 감지 트리거 추가

애플리케이션 > 카탈로그 > 목록

← 트리거 목록
트리거 생성

예약 액션
 빌드 후 액션

이름

트리거 이름을 입력해 주세요.

스케줄

스케줄을 입력해 주세요. ▾



버전관리 시스템 저장소 변경 감지



컨테이너 이미지 변경 감지



http 요청에 대한 응답 감지



정기적으로 트리거 실행

트리거 수행 서비스 어카운트

▾

버전관리 시스템의 저장소 인증 시크릿

▾

✕

vcs 카테고리 태스크 이름

▾

✕

+ 옵션 추가하기

사용 되는 키는 다음과 같다

키	설명
트리거 수행 서비스 어카운트	필수 벨류 빌드(배포)를 수행할 인증 서비스 어카운트 권한이 없다면 빌드(배포)가 실패한다.
버전관리 시스템의 저장소 인증 시크릿	vcs 저장소에 대한 인증 시크릿 Task 에 해당하는 태스크 사양(spec)에서 values.auth.cred 값이 있다면 자동 할당 지원되는 시크릿 타입 : kubernetes.io/basic-auth , kubernetes.io/ssh-auth
vcs 카테고리 태스크 이름	카탈로그/파이프라인 사양(spec)에서 기술된 "vcs 카테고리 태스크" 로 자동 할당 (권장)

컨테이너 이미지 변경 감지 트리거 내부 이미지

애플리케이션 > 카탈로그 > 목록

← 트리거 목록

트리거 생성

예약 액션 빌드 후 액션

이름

스케줄



버전관리 시스템 저장소 변경 감지



컨테이너 이미지 변경 감지



http 요청에 대한 응답 감지



정기적으로 트리거 실행

<input type="text" value="트리거 수행 서비스 어카운트"/>	<input type="text" value=""/>
<input type="text" value="이미지 카테고리 태스크 이름"/>	<input type="text" value=""/>
<input type="text" value="이미지 메타정보 대상 종류"/>	<input type="text" value=""/>
<input type="text" value="이미지 메타정보 (카탈로그/파이프라인)대상 지정"/>	<input type="text" value="이미지 메타정보 대상 종류 설정이 필요합니다."/>
<input type="text" value="트리거 동작 모드"/>	<input type="text" value=""/>
<input type="text" value="컨테이너 이미지 내부/외부 설정"/>	<input type="text" value="내부 컨테이너 이미지"/>
<input type="text" value="레지스트리 저장소에 대한 인증 시크릿"/>	<input type="text" value=""/>

+ 옵션 추가하기

키	설명
트리거 수행 서비스 어카운트	필수 벨류 빌드(배포)를 수행할 인증 서비스 어카운트 권한이 없다면 빌드(배포)가 실패한다.
이미지 카테고리 태스크 이름	이미지 카테고리 태스크 이름 카탈로그/파이프라인 사양(spec)에서 기술된 "이미지 카테고리 태스크" 로 자동 할당 (권장)
이미지 메타정보 대상 종류	카탈로그 또는 파이프라인을 지정할 수 있다.
이미지 메타정보 (카탈로그/파이프라인)대상 지정	이미지 메타데이터를 가져올 카탈로그/파이프라인 대상

키	설명
트리거 동작 모드	<p>digest : 메타데이터의 다이제스트와 레지스트리를 비교하여 변경 여부 확인</p> <p>sequence : 태그 버전 비교 방식으로 레지스트리의 마지막 버전 태그가 메타데이터의 버전 태그보다 큰 지 확인</p> <p>기본값 : 메타데이터의 이미지 태그가 버전방식이면 sequence 아니면 digest 로 자동 할당</p> <p>기본값은 트리거를 등록하는 대상</p> <p>네임스페이스를 선택 하고 해당 네임스페이스의 카탈로그/파이프라인 대상을 선택할 수 있다.</p>
컨테이너 이미지 내부/외부 설정	감지할 이미지 대상의 종류를 설정한다.
레지스트리 저장소에 대한 인증 시크릿	<p>레지스트리 저장소에 대한 인증 시크릿</p> <p>"task"에 해당하는 태스크 사양(spec)에서 values.auth.cred 값을 우선 찾고, 없다면 IMAGE.REGISTRYNAME 환경변수를 자동할당</p> <p>지원되는 시크릿 타입 : kubernetes.io/basic-auth , kubernetes.io/dockerconfigjson</p>

외부 이미지

애플리케이션 > 카탈로그 > 목록

← 트리거 목록
트리거 생성

예약 액션
 빌드 후 액션

이름

트리거 이름을 입력해 주세요.

스케줄

스케줄을 입력해 주세요.



버전관리 시스템 저장소 변경 감지



컨테이너 이미지 변경 감지



http 요청에 대한 응답 감지



정기적으로 트리거 실행

트리거 수행 서비스 어카운트

▼

컨테이너 이미지 내부/외부 설정

외부 컨테이너 이미지 ✕

+ 옵션 추가하기

키	설명
트리거 수행 서비스 어카운트	<p>필수 밸류</p> <p>빌드(배포)를 수행할 인증 서비스 어카운트</p> <p>권한이 없다면 빌드(배포)가 실패한다.</p>
컨테이너 이미지 내부/외부 설정	감지할 이미지 대상의 종류를 설정한다.

http 요청에 대한 응답 감지 트리

예약 액션 빌드 후 액션

이름

스케줄



버전관리 시스템 저장소 변경 감지



컨테이너 이미지 변경 감지



http 요청에 대한 응답 감지



정기적으로 트리거 실행

<input type="text" value="트리거 수행 서비스 어카운트"/>	<input type="text" value=""/>
<input type="text" value="요청대상이 되는 url"/>	<input type="text" value="http://httpbin.org/json"/>
<input type="text" value="응답 데이터와 비교할 데이터"/>	<input type="text" value="Sample Slide Show"/>
<input type="text" value="응답 데이터 필터링"/>	<input type="text" value=".slideshow.title"/> ✕
<input type="text" value="요청에 포함되는 json직렬화 형식의 데이터"/>	<input type="text" value='{"test": "data"}'/> ✕
<input type="text" value="요청에 포함되는 메소드"/>	<input type="text" value="GET"/> ✕
<input type="text" value="요청에 포함되는 header 목록"/>	<input type="text" value="useragent"/> <input type="text" value="window"/> ✕
	<input type="text" value="cookie"/> <input type="text" value="cookie-value"/>

+ 옵션 추가하기

키	설명
트리거 수행 서비스 어카운트	필수 밸류 빌드(배포)를 수행할 인증 서비스 어카운트 권한이 없다면 빌드(배포)가 실패한다.
요청대상이 되는 url	필수 밸류
응답 데이터와 비교할 데이터	필수 밸류 응답 데이터와 비교할 데이터로 단순 문자열 또는 json직렬화 형식
응답 데이터 필터링	응답 데이터를 필터링하여 응답 데이터와 비교할 데이터 와의 값을 비교 jsonpath 형식
요청에 포함되는 json직렬화 형식의 데이터	request body 에 포함되는 json 데이터
요청에 포함되는 메소드	http 메소드 기본값 : GET
요청에 포함되는 header 목록	http 요청에 사용되는 헤더를 여러개 등록할 수 있다.

정기적으로 트리거 실행

애플리케이션 > 키달로그 > 목록

← 트리거 목록

트리거 생성

예약 액션 빌드 후 액션

이름

스케줄
 ▾



버전관리 시스템 저장소 변경 감지



컨테이너 이미지 변경 감지



http 요청에 대한 응답 감지



정기적으로 트리거 실행

트리거 수행 서비스 어카운트
 ▾

+ 옵션 추가하기

키	설명
트리거 수행 서비스 어카운트	필수 벨류 빌드(배포)를 수행할 인증 서비스 어카운트 권한이 없다면 빌드(배포)가 실패한다.

카탈로그/파이프라인 빌드(배포) 트리거

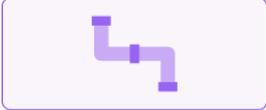
애플리케이션 > 카탈로그 > 목록

← 트리거 목록

트리거 생성

예약 액션 빌드 후 액션

이름



카탈로그/파이프라인 빌드 트리거



웹훅 전달 트리거

트리거 수행 서비스 어카운트

대상(카탈로그/파이프라인) 종류

빌드(배포)를 수행할 대상 지정

트리거 수행 매칭 조건 빌드 상태 목록

+ 옵션 추가하기

메타데이터	예시
트리거 수행 서비스 어카운트	필수 벨류 빌드(배포)를 수행할 인증 서비스 어카운트 권한이 없다면 빌드(배포)가 실패한다.
대상(카탈로그/파이프라인) 종류	필수 벨류 빌드(배포) 대상의 종류 카탈로그 또는 파이프라인을 지정할 수 있다.
빌드(배포)를 수행할 대상 지정	필수 벨류 빌드(배포) 대상 네임스페이스를 선택 하고 해당 네임스페이스의 카탈로그/파이프라인 대상을 선택할 수 있다.
트리거 수행 매칭 조건 빌드 상태 목록	사용가능한 phase '성공, 실패, 에러, 종료' 빌드 phase 결과와 매칭되는 경우에만 트리거 수행 기본값 : Succeeded

웹훅 전달 트리거

애플리케이션 > 카탈로그 > 목록

← 트리거 목록

트리거 생성

예약 액션 빌드 후 액션

이름

트리거 이름을 입력해 주세요.



카탈로그/파이프라인 빌드 트리거



웹훅 전달 트리거

웹훅을 전달할 대상 url http://localhost:8080/json

타임아웃(기본값: 5s) 1h15m30s ✕

트리거 수행 매칭 조건 빌드 상태 목록
성공 ✕
실패 ✕
에러 ✕
종료 ✕
▼ ✕

+ 옵션 추가하기

메타데이터	예시
웹훅을 전달할 대상 url	필수 벨류 웹훅 호스트
타임아웃(기본값: 5s)	webhook 전달시 timeout 으로 사용된다. Duration 타입
트리거 수행 매칭 조건 빌드 상태 목록	사용가능한 phase '성공, 실패, 에러, 종료' 빌드 phase 결과와 매칭되는 경우에만 트리거 수행 기본값 : Succeeded

트리거 수행 서비스 어카운트 생성

CAUTION 클러스터 를 권한이 없는 사용자인 경우 빌드(배포)를 수행할 대상의 네임스페이스는 현재 네임스페이스만 적용가능하다

서비스 어카운트 생성

서비스 어카운트 생성 방법은 해당 가이드를 참고 한다.

- 서비스 어카운트 생성 예시

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: tgr-build-executor
  namespace: test
```

YAML

롤과 롤바인딩 생성

현재 네임스페이스의 빌드(배포) 할 때 사용되는 권한을 설정한다.

롤 과 롤 바인딩 생성 방법은 해당 가이드를 참고 한다.

- 롤 생성 예시
 - 기본으로 빌드(배포) 하는데 사용되는 권한이며 빌드(배포) 리소스에 대한 권한이 추가적으로 필요한 경우 기본으로 사용되는 권한과 필요한 권한을 추가하여 사용한다.

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: tgr-build-executor-role
  namespace: test
rules:
  - verbs:
    - get, list, watch
    apiGroups:
    - cicd.accordions.co.kr
    resources:
    - pipelines, catalogs
  - verbs:
    - create
    apiGroups:
    - cicd.accordions.co.kr
    resources:
    - buildrequests
```

YAML

• 를 바인딩 생성 예시

YAML

```

kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: tgr-build-executor-rolebinding
  namespace: test
subjects:
  - kind: ServiceAccount
    name: tgr-build-executor
    namespace: test
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: tgr-build-executor-role

```

클러스터 룰과 클러스터 룰 바인딩 생성

클러스터 룰 권한이 있는 경우 사용 가능하며 다른 네임스페이스의 빌드(배포) 할 때 사용되는 권한을 설정한다.

클러스터 룰, 클러스터 룰 바인딩 생성 방법은 해당 가이드를 참고 한다.

• 클러스터 룰 생성 예시

- 기본으로 빌드(배포) 하는데 사용되는 권한이며, 빌드(배포) 리소스에 대한 권한이 추가적으로 필요한 경우 기본으로 사용되는 권한과 필요한 권한을 추가하여 사용한다.

YAML

```

kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: tgr-build-executor-clusterrole
rules:
  - verbs:
    - get, list, watch
    apiGroups:
    - cicd.accordions.co.kr
    resources:
    - pipelines, catalogs
  - verbs:
    - create
    apiGroups:
    - cicd.accordions.co.kr
    resources:
    - buildrequests

```

• 클러스터 룰 바인딩 생성 예시

YAML

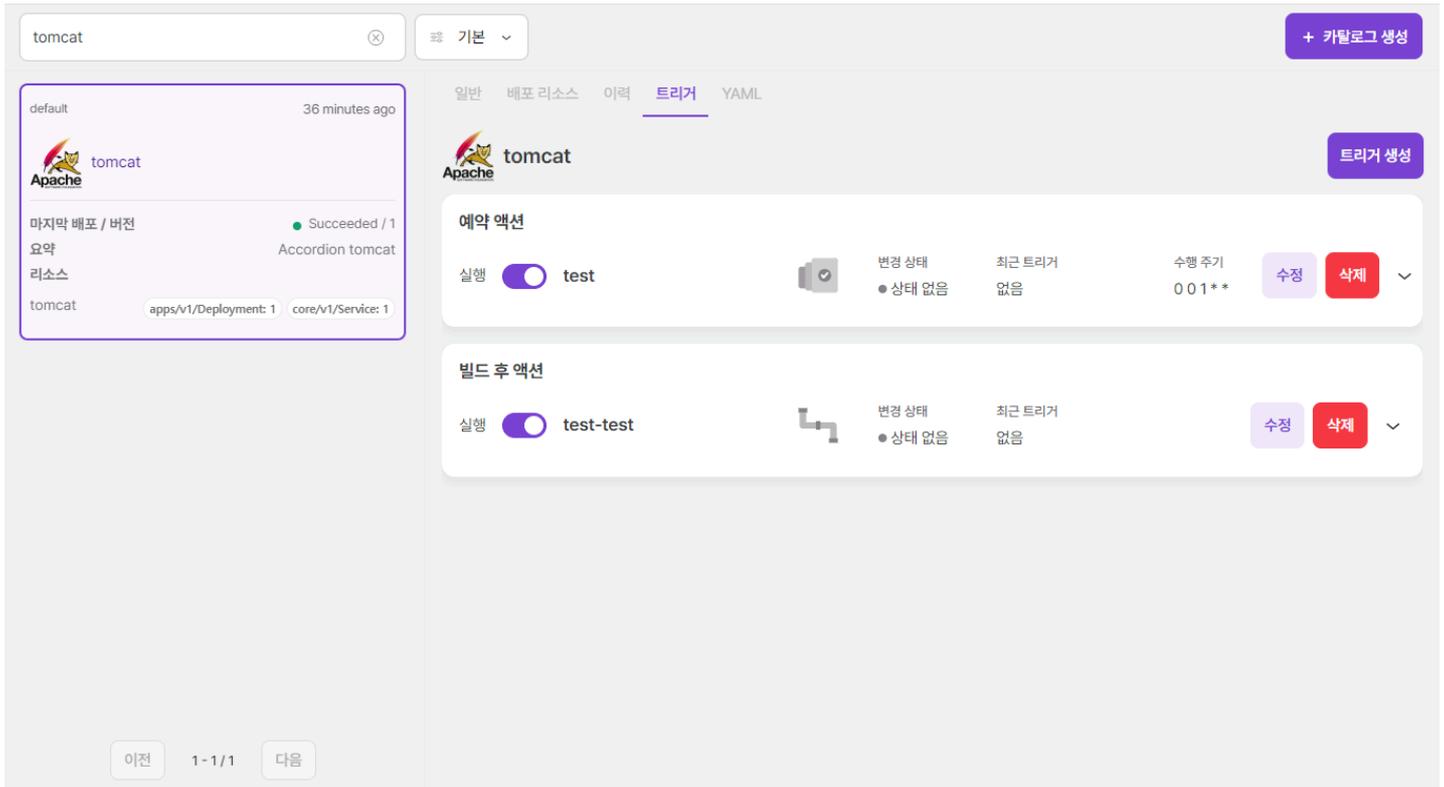
```

kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: tgr-build-executor-clusterrolebinding
subjects:
  - kind: ServiceAccount
    name: tgr-build-executor
    namespace: test
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: tgr-build-executor-clusterrole

```

트리거 목록

애플리케이션 > 카탈로그 > 목록



해당 카탈로그 또는 파이프라인에 등록된 트리거들의 목록을 확인할 수 있다.

실행 토글 버튼은 해당 트리거의 실행 여부를 설정하는 버튼이며 보라색으로 설정된 경우 동작하며, 회색으로 변경된 경우 동작하지 않는다.

변경 상태 는 해당 트리거의 상태 정보이며 예약 액션 트리거 의 경우 성공 과 실패 가 있으며, 빌드 후 액션 트리거 의 경우 성공 , 실패 , 스킵 이 존재한다.

종류	설명
성공	트리거가 감지하여 작업을 수행한 경우
실패	트리거가 작업도중 실패한 경우
스킵	감지는 했으나 조건이 일치하지 않은 경우, 이미 빌드 중인 경우

최근 트리거 는 해당 트리거의 최근 동작 시간이며 예약 액션 트리거 의 경우 성공 또는 실패 시간만 나타낸다.

수행 주기 는 예약 액션 트리거 에서만 볼수 있고 해당 트리거의 스케줄을 나타낸다.

트리거 상세

실행 test-manual HTTP 변경 상태 ● 실패 최근 트리거 2023.09.18 09:42:01 수행 주기 0 0 1 ** 수정 삭제 ^

메시지

```
[.deploy.volumes[0].from.name is required, .deploy.volumes[0].from.type invalid enum (allowed: [configMap]), .deploy.volumes[0].from.volumeMounts invalid "array" type]
[.deploy.volumes[0].from.type invalid enum (allowed: [secret]), .deploy.volumes[0].from.volumeMounts invalid "array" type, .deploy.volumes[0].from.name is required]
[.deploy.volumes[0].from.name is required, .deploy.volumes[0].from.type invalid enum (allowed: [persistentVolumeClaim]), .deploy.volumes[0].from.volumeMounts invalid "array" type].deploy.volumes[0].from.volumeMounts invalid "array" type], .deploy.image is required]
```

조건 감지	최근 체크 시간	키	밸류
● 실패	2023.09.18 09:42:01	요청에 포함되는 json직렬화 형식의 데이터	data
		응답 데이터 필터링	url
		요청에 포함되는 header 목록	useragent : tst
		요청에 포함되는 메소드	method
		트리거 수행 서비스 어카운트	tgr-builder
		요청대상이 되는 url	http://httpbin.org/get
		응답 데이터와 비교할 데이터	http://httpbin.org/get

원하는 트리거를 클릭 하면 상세 정보를 확인할 수 있다.

예약 액션 트리거 의 경우, 상단에서 실행 여부, 상태 메세지 를 확인할 수 있고 하단 표에서 조건 감지, 최근 체크시간, 키, 밸류 를 확인할 수 있다.

트리거의 조건 감지 종류는 아래와 같다.

조건 감지 는 예약 액션 트리거 의 트리거의 상태 정보 와 동일 하며 미감지 와 스킵 이 추가 된다.

종류	설명
미감지	예약 액션 트리거 에서만 사용되며 트리거가 감지되지 않은 경우

최근 체크 시간 은 해당 트리거의 스케줄에 따라 주기적으로 감지를 시도한 시간이다.

빌드 후 액션

실행 after-build 🔗 변경 상태 ● 실패 최근 트리거 2023.09.19 11:23:22 수정 삭제 ^

메시지

```
Failed trigger for build : Pipeline.cicd.accordions.co.kr "test" not found
```

키	밸류
대상(카탈로그/파이프라인) 종류	Pipeline
트리거 수행 서비스 어카운트	tgr-builder
빌드(배포)를 수행할 대상 지정	manual/test

빌드 후 액션 트리거 의, 경우 상단에서 실행 여부, 상태 메세지 를 확인할 수 있고 하단 표에서 키, 밸류 를 확인할 수 있다.

트리거 수정

애플리케이션 > 카탈로그 > 목록

< 트리거 목록

트리거 수정

예약 액션 빌드 후 액션

이름
schedule-trigger

스케줄
0 0 1 * * ▼

버전관리 시스템 저장소 변경 감지

컨테이너 이미지 변경 감지

http 요청에 대한 응답 감지

정기적으로 트리거 실행

트리거 수행 서비스 어카운트
tgr-builder ▼

+ 옵션 추가하기

트리거 우측의 수정 버튼을 클릭 하면 수정을 할 수 있는 상태가 된다.

옵션 추가하기 버튼을 클릭하여 새로운 옵션을 입력할 수 있다.

수정이 완료가 되면 우측 상단의 트리거 수정 버튼을 클릭하여 저장할 수 있고 또는 트리거 목록 버튼을 클릭하여 수정을 취소할 수 있다.

트리거 삭제

애플리케이션 > 카탈로그 > 목록

tomcat ⊗ 기본 ▼ + 카탈로그 생성

default 40 minutes ago

마지막 배포 / 버전
요약
리소스
tomcat

일반 배포 리소스 이력 트리거 YAML

예약 액션

변경 상태	최근 트리거	수행 주기	수정	삭제
● 상태 없음	없음	0 0 1 * *	수정	삭제 ▼

정말로 삭제하시겠습니까?
확인을 위해 tomcat/test를 입력해 주세요.

닫기
삭제하기

삭제를 원하는 트리거의 우측 삭제 버튼을 클릭하면 모달이 나타나는데 모달에서 네임스페이스와 카탈로그 또는 파이프라인의 이름을 입력 후 삭제한다.

트리거 이벤트

트리거가 실제 수행되는 경우, 성공/실패에 대한 내용을 kubernetes event를 통하여 등록된다.

아래는 이벤트 예시이다.

```

49m Normal SucceededTrigger pipeline/hello Succeeded trigger for "build" : pipeline/default/image
48m Normal SucceededTrigger pipeline/hello Succeeded trigger for "request" : GET 'http://httpbin.org/ip'
50m Warning FailedTrigger pipeline/hello Failed trigger for "build" : xxxx

```

빌드 보관 정책

빌드시 늘어나는 빌드의 개수를 조절하기 위한 정책으로 조건에 맞는 빌드만 보관하고 나머지는 삭제하는 정책이다.

카탈로그/파이프라인의 다음과 같은 어노테이션으로 동작한다.

어노테이션	설명
cidc.accordions.co.kr/retention.count	개수로 보관할 빌드의 개수를 정한다.
cidc.accordions.co.kr/retention.period	기간으로 보관할 빌드의 기간을 정한다.

등록

```

metadata:
  annotations:
    cidc.accordions.co.kr/retention.count: '1'
    cidc.accordions.co.kr/retention.period: 1m

```

YAML

빌드보관 정책은 다중 등록시 count 가 우선 적용되며 **1개의 빌드를 최소 보관한다.**

타입	입력 가능한 값	설명
count	Number	1 = 1개, 2 = 2개, 10 = 10개
period	Duration	1d = 1일, 1h = 1시간, 1m = 1분, 1s = 1초, 2d2h = 2일 2시간

count

count 가 **1인 경우 1개의** 최신 빌드만 보관한다.

period

period 가 **1m** 인 경우 **1m + 마지막 빌드시간이 현재시간 이후** 인 빌드만 보관한다.

프로세스

빌드보관 정책은 background 와 buildRequest 삭제 이벤트 발생을 기준으로 각각 동작한다.

count 또는 period annotation 이 존재하는 경우 삭제가 진행된다.

정상적으로 삭제되는 경우 카탈로그는 buildStatus.lastDeletedVersion 에 등록되며,

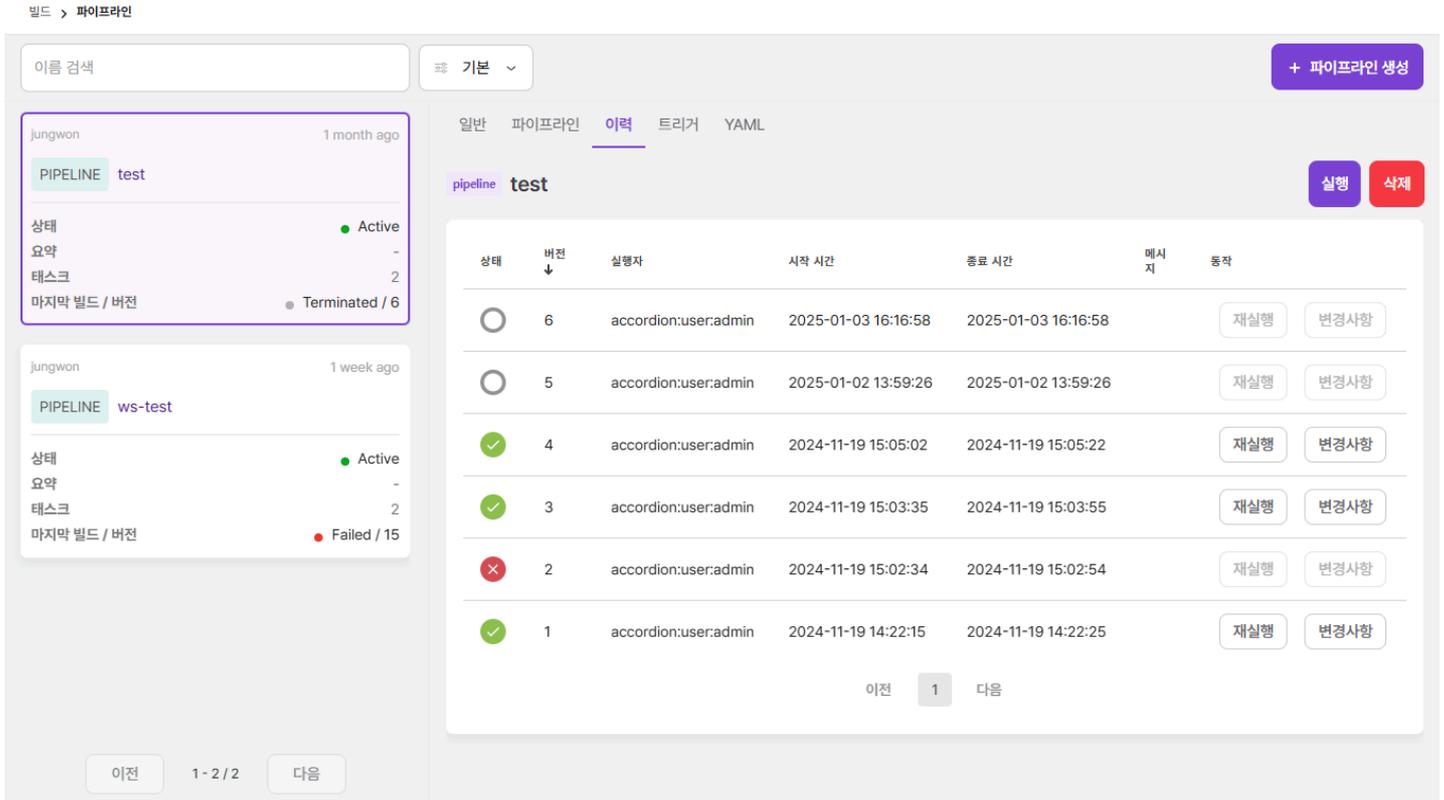
파이프라인은 status.lastDeletedVersion 에 등록된다.

4.3.3. 빌드

빌드에서는 파이프라인, 승인, 태스크 템플릿 등의 메뉴를 이용해 파이프라인 또는 카탈로그와 파이프라인에서 사용할 태스크 템플릿을 작성하고 파이프라인에서 요구하는 승인 절차를 처리할 수 있다.

4.3.3.1. 파이프라인

파이프라인은 이미지 빌드 및 배포를 위해 한개 이상의 태스크를 가지고 태스크의 연관 관계(디펜던시)를 관리한다. 파이프라인은 지속적으로 수행이 가능하며 수행에 대한 이력을 관리할 수 있다.

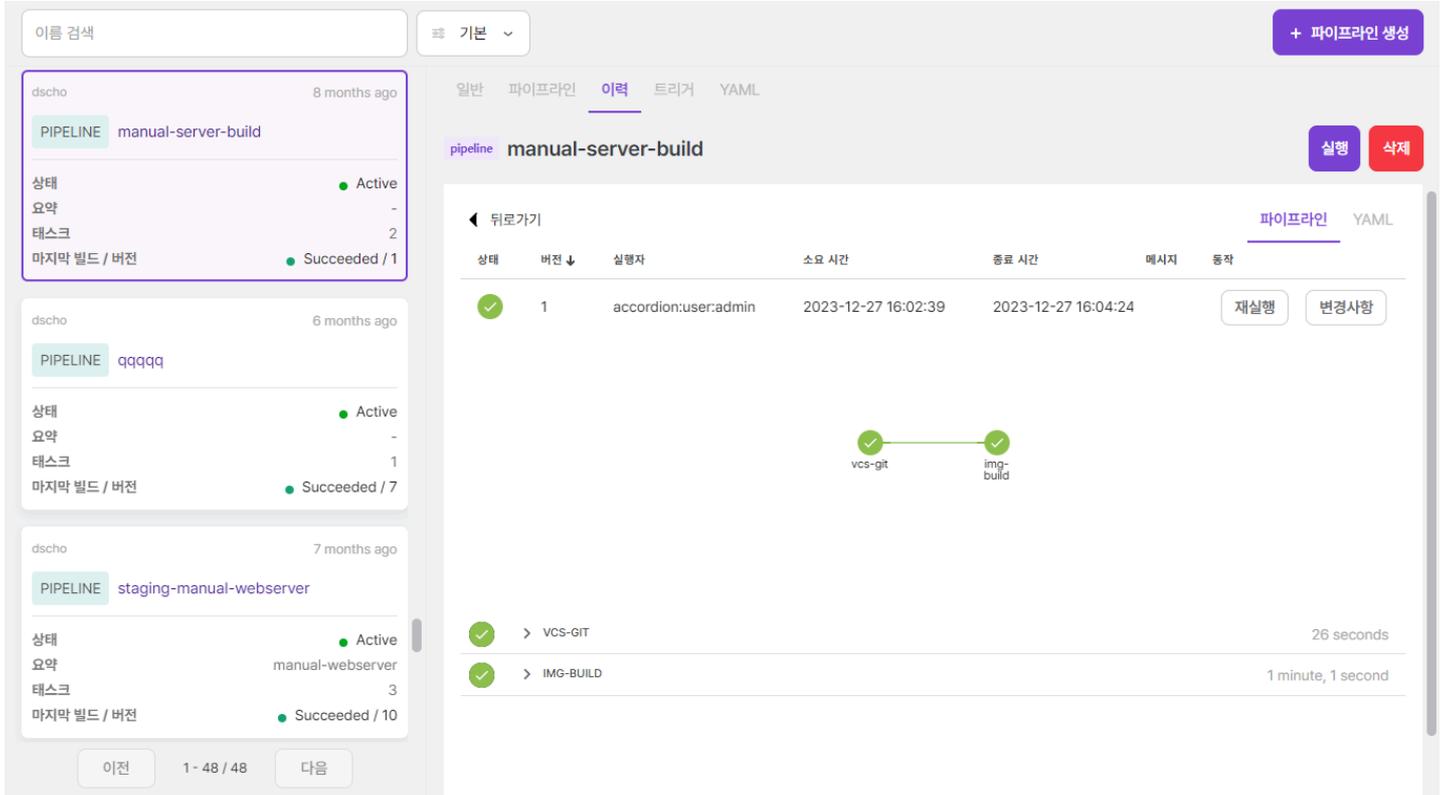


파이프라인 메뉴에서는 파이프라인을 실행한 빌드 이력을 확인할 수 있다. 이때 개별 빌드에 대한 상태 정보가 아이콘으로 표시되는데 각각이 의미하는 바는 다음과 같다.

이미지	태스크 상태
	종료(Terminated): 사용자가 빌드를 중지한 상태
	성공(Succeeded) : 태스크가 수행하고 종료코드(exit code)가 0으로 완료된 상태
	실행 중(Running) : 태스크가 수행되고 있는 상태
	실패(Failed) : 태스크가 수행되었지만 종료코드가 0이 아닌 것으로 완료된 상태
	에러(Error) : 태스크가 실행되지 못하거나 비정상 종료된 상태 발생 예시 <ul style="list-style-type: none"> • 쿠버네티스 인프라적인 요인으로 태스크가 수행되지 않는 경우 • 수행 중인 태스크가 외부 요인에 의해서 비정상 종료되는 경우

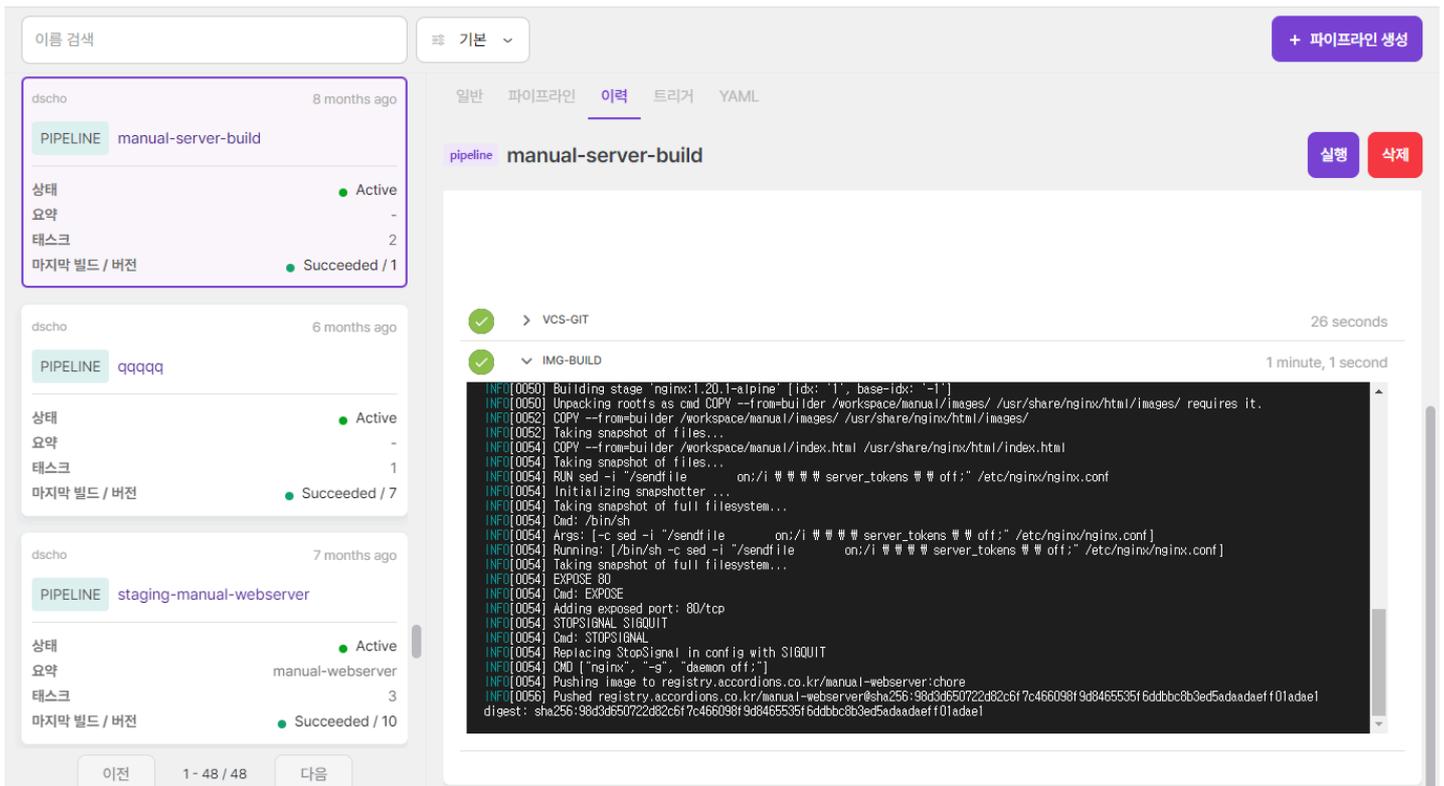
이미지	태스크 상태
	대기(Pending) : 태스크가 수행 전 대기하는 상태
	경고(Warning): 태스크를 수정 중에 잘못된 데이터를 입력하는 경우 발생

빌드 이력을 선택하면 개별 빌드에 대한 상세 정보를 확인할 수 있다.



The screenshot shows the 'manual-server-build' pipeline details. On the left, a list of pipelines is shown, with 'manual-server-build' selected. The main area displays the pipeline's execution history, including a table with columns for status, version, executor, start time, end time, and message. Below the table, a progress bar shows the completion of 'vcs-git' and 'img-build' tasks. A detailed log for the 'IMG-BUILD' task is visible at the bottom, showing the build process for nginx:1.20.1-alpine.

개별 빌드는 하나 이상의 태스크로 구성되며 태스크를 선택하면 해당 태스크에서 발생한 로그를 확인할 수 있다.



This screenshot shows the detailed log for the 'IMG-BUILD' task. The log output includes the following information:

```

INFO[0050] Building stage 'nginx:1.20.1-alpine' [idx: '1', base-idx: '-1']
INFO[0050] Unpacking roots as cid COPY --from=builder /workspace/manual/images/ /usr/share/nginx/html/images/ requires it.
INFO[0052] COPY --from=builder /workspace/manual/images/ /usr/share/nginx/html/images/
INFO[0052] Taking snapshot of files...
INFO[0054] COPY --from=builder /workspace/manual/index.html /usr/share/nginx/html/index.html
INFO[0054] Taking snapshot of files...
INFO[0054] RUN sed -i '/sendfile on;/i \#\#\#\# server_tokens \#\#\# off;' /etc/nginx/nginx.conf
INFO[0054] Initializing snapshotter ...
INFO[0054] Taking snapshot of full filesystem...
INFO[0054] Cmd: /bin/sh
INFO[0054] Args: [-c sed -i '/sendfile on;/i \#\#\#\# server_tokens \#\#\# off;' /etc/nginx/nginx.conf]
INFO[0054] Running: [/bin/sh -c sed -i '/sendfile on;/i \#\#\#\# server_tokens \#\#\# off;' /etc/nginx/nginx.conf]
INFO[0054] Taking snapshot of full filesystem...
INFO[0054] EXPOSE 80
INFO[0054] Cmd: EXPOSE
INFO[0054] Adding exposed port: 80/tcp
INFO[0054] STOPSIGNAL SIGQUIT
INFO[0054] Cmd: STOPSIGNAL
INFO[0054] Replacing StopSignal in config with SIGQUIT
INFO[0054] CMD ["nginx", "-g", "daemon off;"]
INFO[0054] Pushing image to registry.accordions.co.kr/manual-webserver:chore
INFO[0056] Pushed registry.accordions.co.kr/manual-webserver@sha256:98d3d650722d82c6f7c466098f9d8465535f6ddbcb3ed5adaaeff01adaef
digest: sha256:98d3d650722d82c6f7c466098f9d8465535f6ddbcb3ed5adaaeff01adaef
  
```

실패한 태스크의 경우 실패한 원인을 확인할 수 있다.

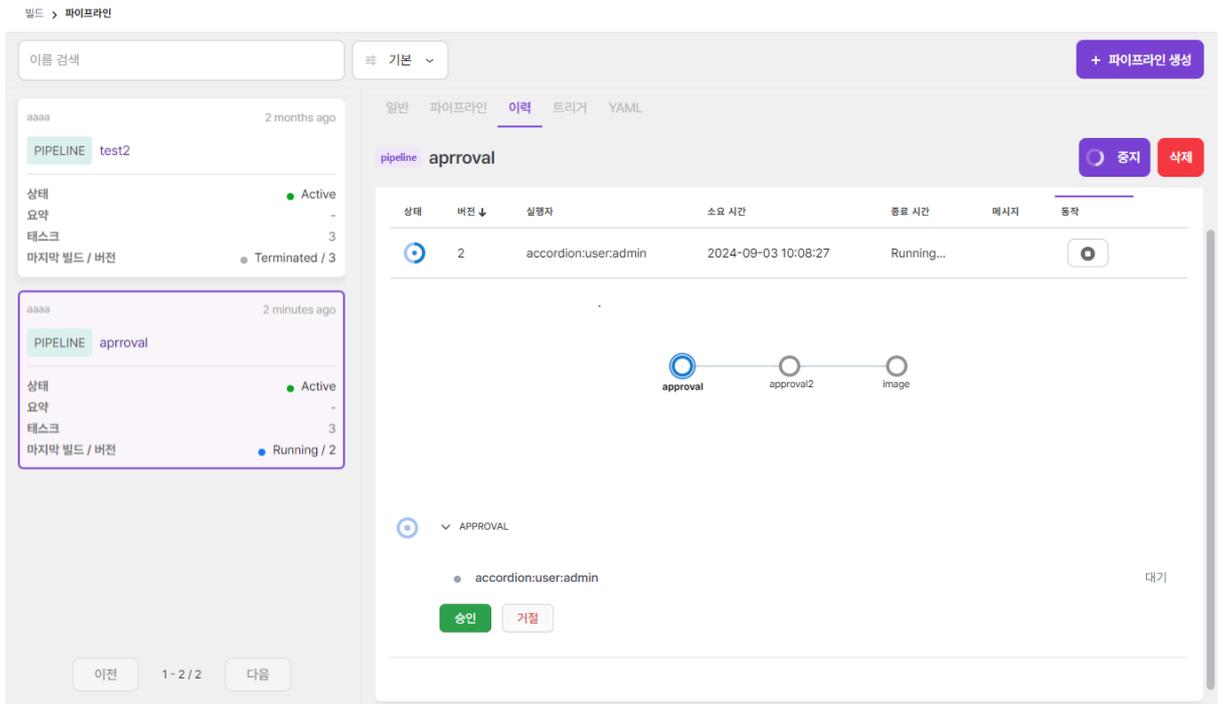
NOTE

빌드 실패 원인



실패
Error (exit code 1)

태스크에는 승인 태스크도 있다. 승인 태스크의 경우 태스크를 선택했을때 로그 정보가 제공되는 것이 아니라 승인 목록이 나타나고 사용자가 승인자 목록에 포함되어 있으면 승인 또는 거절이 가능하다.



NOTE

승인의 경우 모든 승인자가 승인해야 다음 태스크로 넘어가며 한명의 승인자라도 거절하면 파이프라인은 중지된다.

승인 성공

✓ CHECK 42 seconds

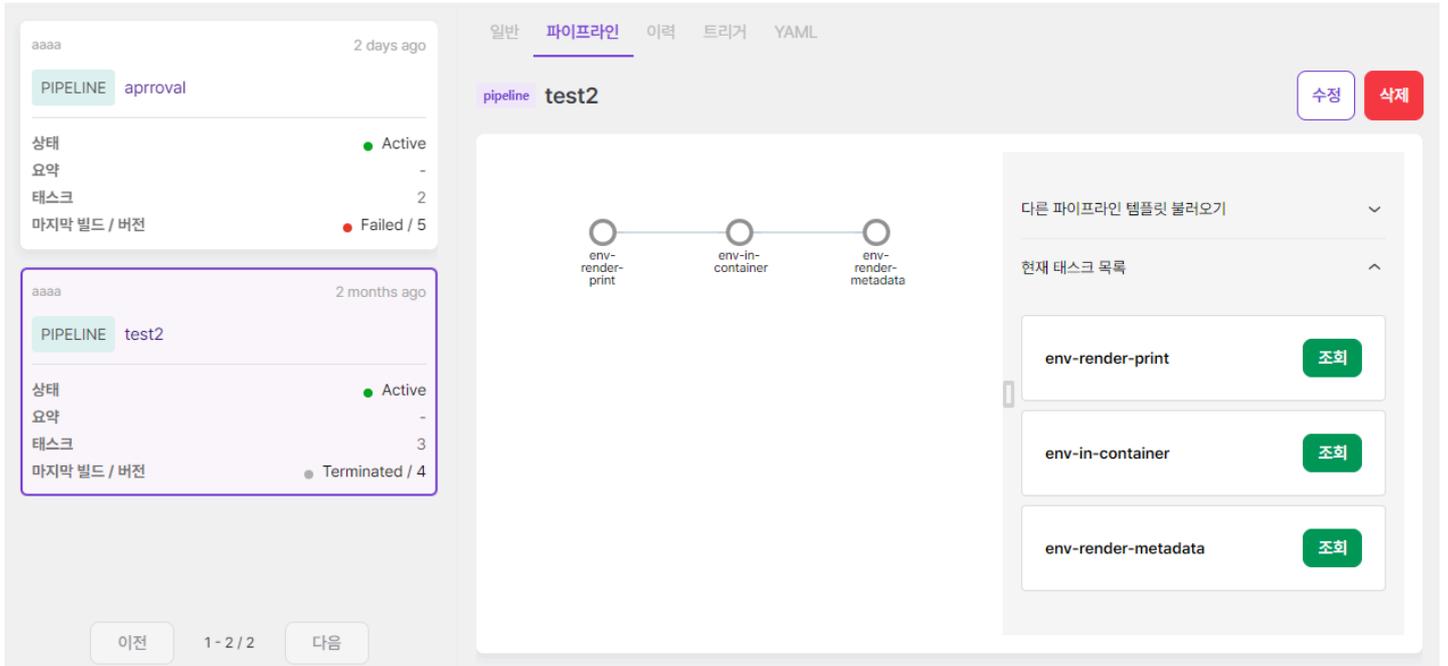
- accordion:user:admin 승인
- accordion:user:tester 승인

승인 실패

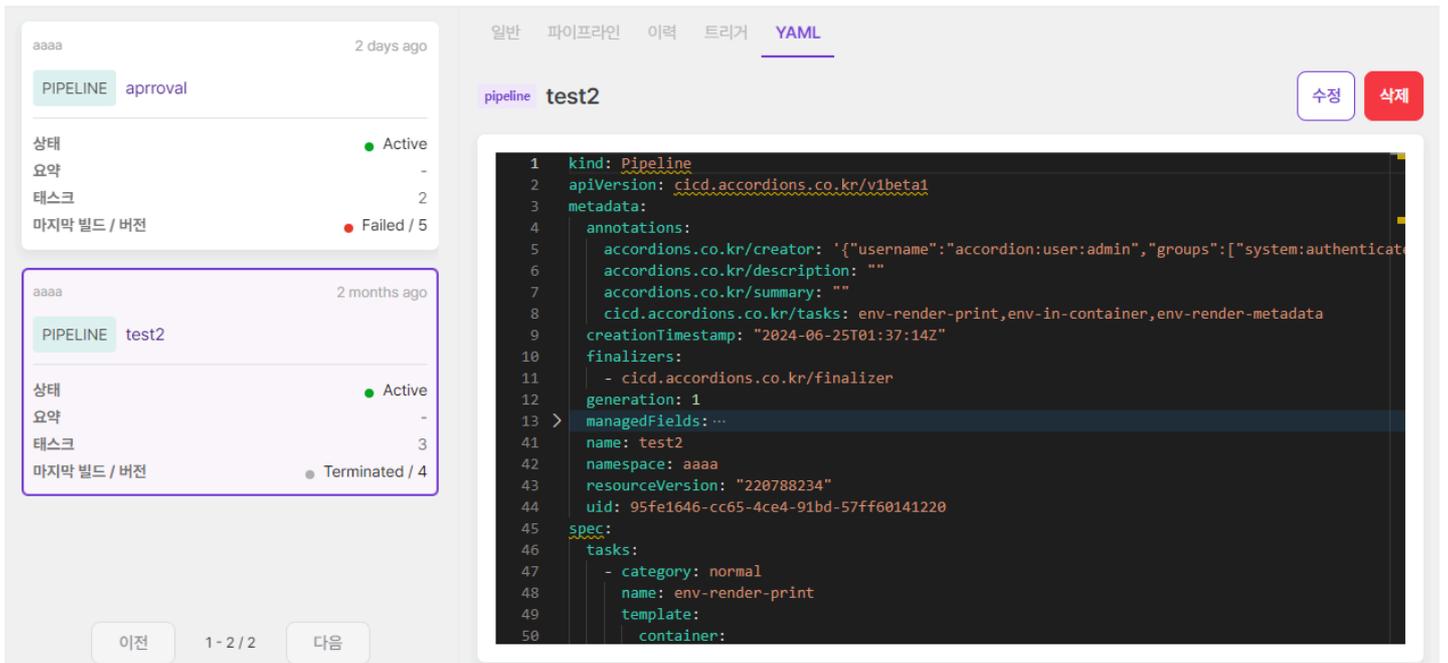
✗ CHECK 8 minutes, 14 seconds

- accordion:user:admin 승인
- accordion:user:tester 거절

상단의 파이프라인 탭을 선택하면 파이프라인을 구성하고 있는 태스크 정보를 그래프로 확인할 수 있다. 그래프의 노드는 태스크를 의미하며 노드를 선택하면 해당 태스크에 대한 정보를 조회할 수 있다.



상단의 YAML 탭을 선택하면 파이프라인을 구성하는 정보를 YAML 편집기로 확인할 수 있으면 UI에서 설정하지 못하는 정보에 대해서도 직접 수정 할 수 있다.



YAML 편집기로 보면 예약된 환경변수를 볼 수 있다. 파이프라인에서 예약된 환경변수는 다음과 같다.

의미	환경변수 표현식
파이프라인 이름	{{{.PIPELINE.NAME}}}
파이프라인 네임스페이스	{{{.PIPELINE.NAMESPACE}}}
파이프라인 UID	{{{.PIPELINE.UID}}}
파이프라인 인스턴스 이름	{{{.PIPELINE.INSTANCE}}}
빌드 이름	{{{.BUILD.NAME}}}

의미	환경변수 표현식
빌드 버전	{{{.BUILD.VERSION}}}
빌드 생성자 이름	{{{.BUILD.CREATOR.USERNAME}}}
빌드 생성자 그룹	{{{.BUILD.CREATOR.GROUPS}}}

파이프라인 생성

+ 파이프라인 생성 버튼을 선택하면 나타나는 화면에서 파이프라인 정보를 입력하여 생성할 수 있다. 생성 시에는 FORM 또는 YAML로 입력할 수 있다.

← 파이프라인 목록 파이프라인 생성

이름

파이프라인 템플릿 선택

빈 템플릿

파이프라인 미리보기

요약

입력하는 정보는 다음과 같다.

항목	설명
파이프라인 이름	생성할 파이프라인 이름
파이프라인 템플릿 선택	파이프라인을 구성하는 태스크에 대한 정보를 템플릿으로 선택
파이프라인 미리보기	파이프라인을 구성하고 있는 태스크 정보를 그래프로 확인
파이프라인 요약	파이프라인에 대한 한줄 요약 (파이프라인 목록에 표시)
파이프라인 설명	파이프라인에 대한 설명을 마크다운으로 작성

파이프라인 생성 시 파이프라인 템플릿에서 태스크에 대한 구성 정보를 불러와 설정할 수 있다. 템플릿을 수정해서 사용하고 싶은 경우 파이프라인 생성 후 수정 화면에서 변경 사항을 반영한다.

Search PipelineTemplate

✓ 빈 템플릿
파이프라인 템플릿을 사용하지 않고, 직접 작성

파이프라인 템플릿

acc-showcase-env
Showcase env

acc-vcs-image
vcs -> image

acc-vcs-jbuild-tomcat
vcs -> java build -> tomcat image

acc-vcs-jbuild-wildfly
vcs -> java build -> wildfly image

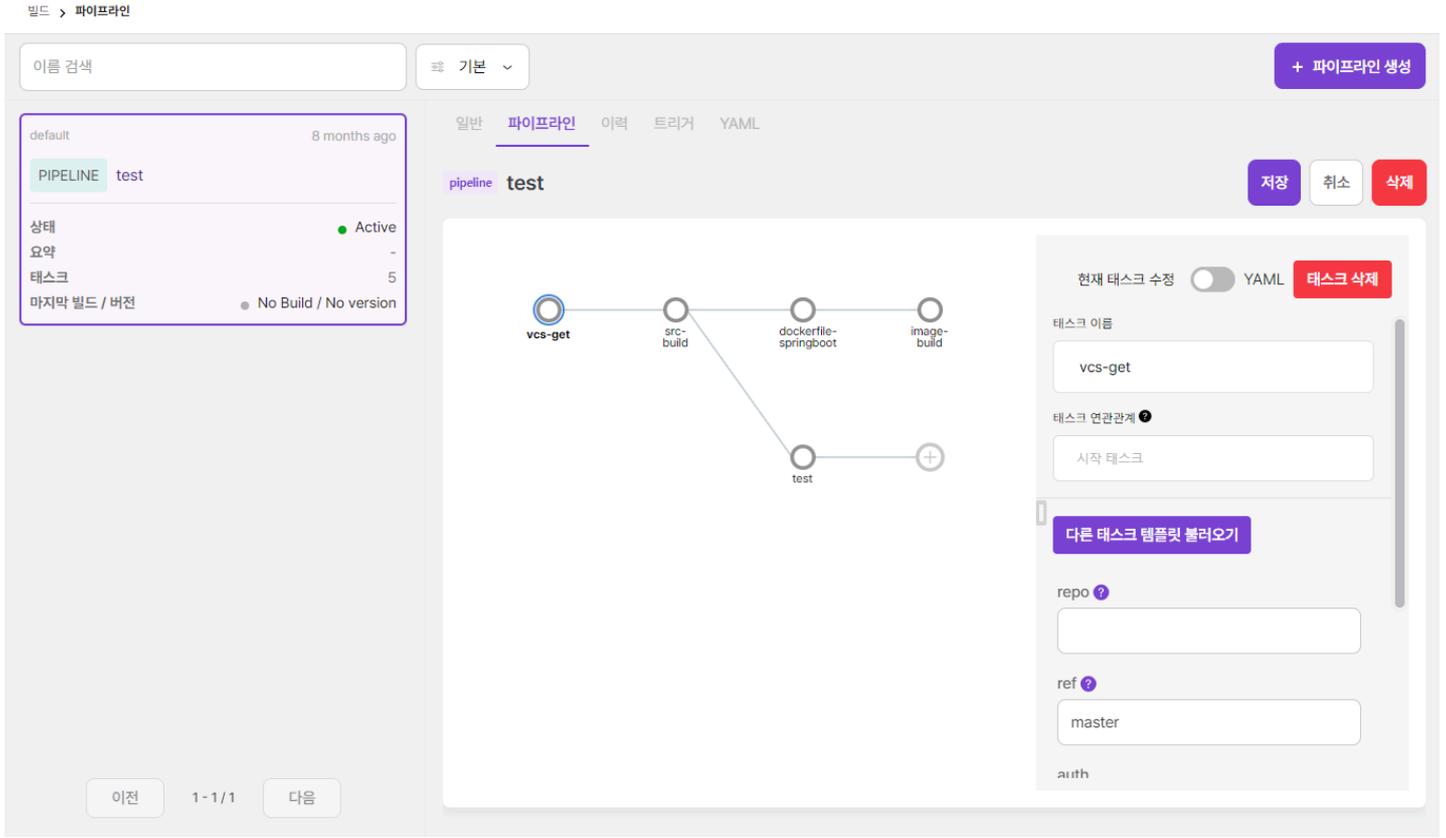
전체 4 중 4

TIP

파이프라인 생성 시 빈 템플릿 을 선택하면 태스크가 없는 파이프라인을 생성할 수 있다. 이 경우 처음부터 파이프라인을 모두 작성할 수 있다.

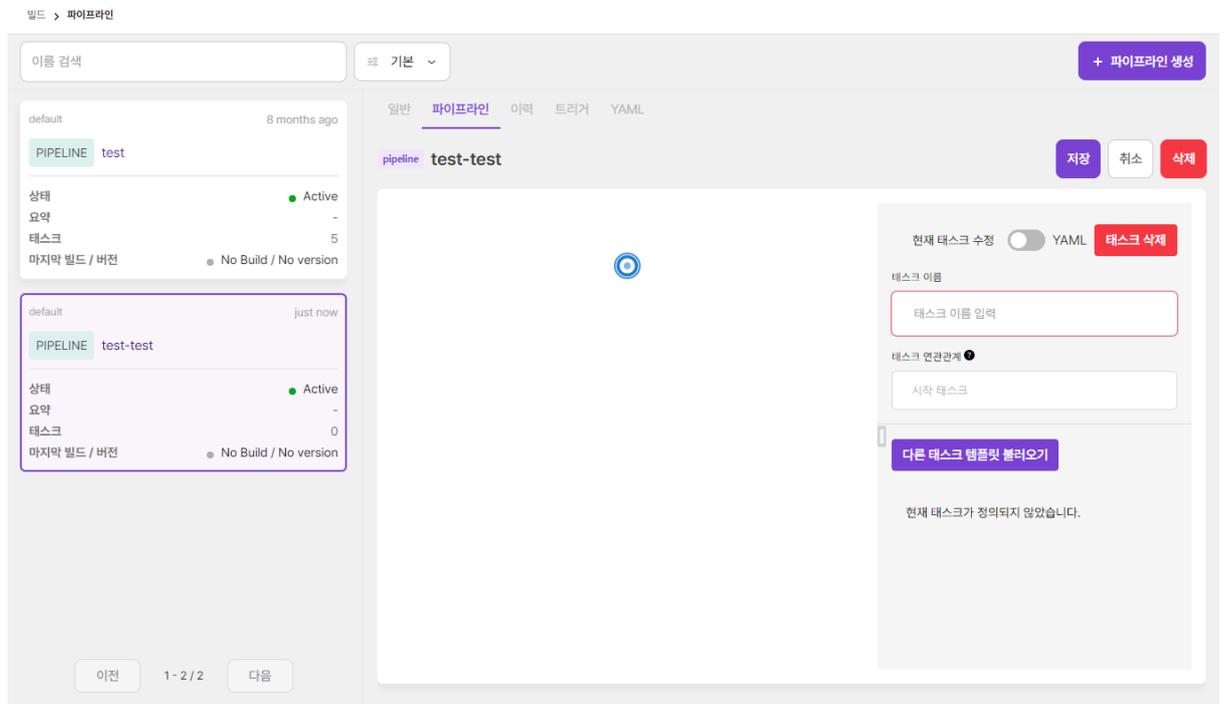
파이프라인 수정

파이프라인을 생성할 때 태스크에 대한 정보는 파이프라인 템플릿에서 가져와 구성한다. 파이프라인의 태스크에 대한 정보를 변경해야하는 경우 변경할 파이프라인을 찾아 파이프라인 탭의 수정 버튼을 선택한다.



태스크가 없는 파이프라인의 경우에는 수정하여, 빈 태스크의 + 버튼을 선택하면 태스크 구성을 할 수 있다.

NOTE



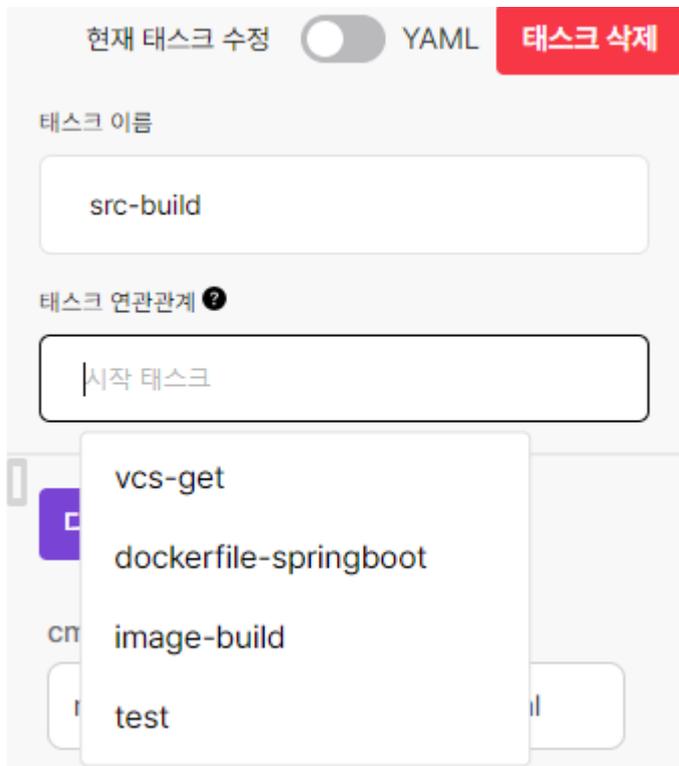
태스크에 대한 스펙을 작성할때에는 이름과 다른 태스크와의 연관관계를 설정하고 상세정보는 태스크 템플릿을 기반으로 작성한다.

태스크의 이름은 쿠버네티스 이름 정책에 맞춰 작성하고 태스크의 연관관계는 해당 태스크를 수행하는 조건에 대한 정보를 입력한다. 이는 {이름}. {상태} 의 형식으로 입력할 수 있다. 상태에 입력할 수 있는 값은 다음과 같다.

상태	설명
Succeeded	성공
Failed	실패
Skipped	생략
Error	에러

연관관계 작성 방법은 다음과 같다.

- 태스크 연결
 - 태스크 연관관계의 입력창을 선택한 후 태스크 설정



- 태스크 상태, 조건 추가
 - 순차적으로 태스크의 상태와 조건을 선택

현재 태스크 수정 YAML **태스크 삭제**

태스크 이름

src-build

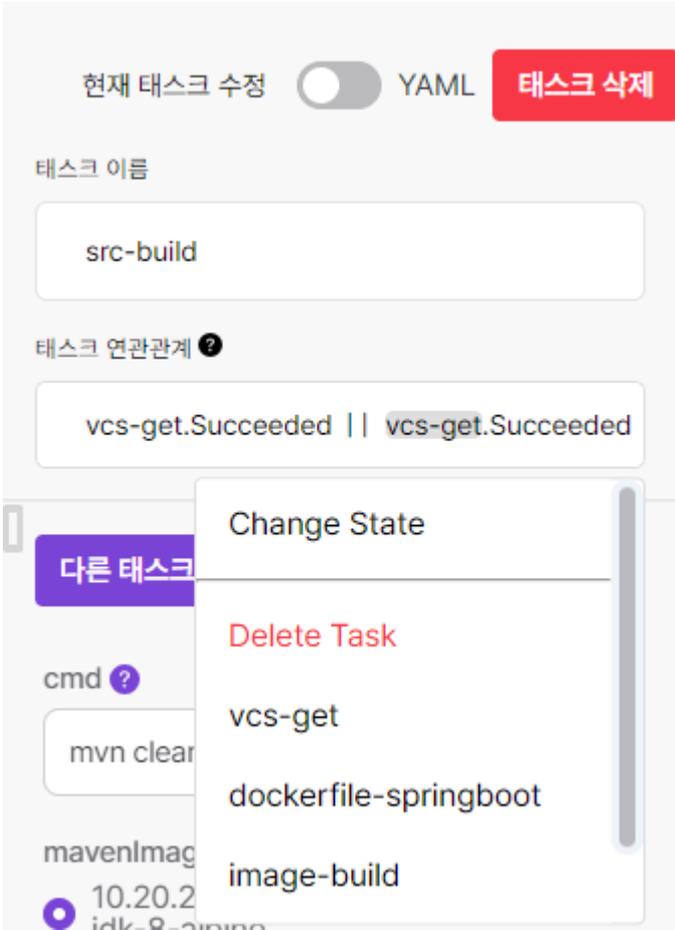
○ 태스크 연관관계 [?]

vcs-get.Succeeded|

다른 태스크

- &&
- ||

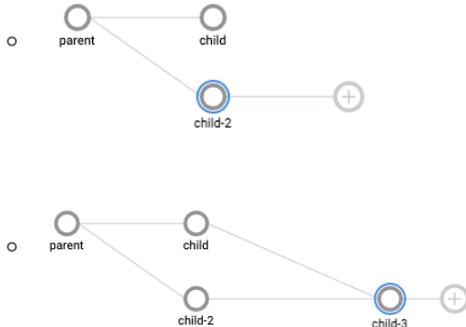
- 테스트 삭제, 수정
 - 수정이 필요한 테스트, 상태를 클릭하여 내용 변경



연관관계 작성의 예는 다음과 같다.

- 조건
 - parent 테스트 후 child 와 child-2 테스트를 수행한다.
 - child 테스트가 성공하고 child-2 테스트가 실패하면 child-3 테스트를 수행한다.
- 설정

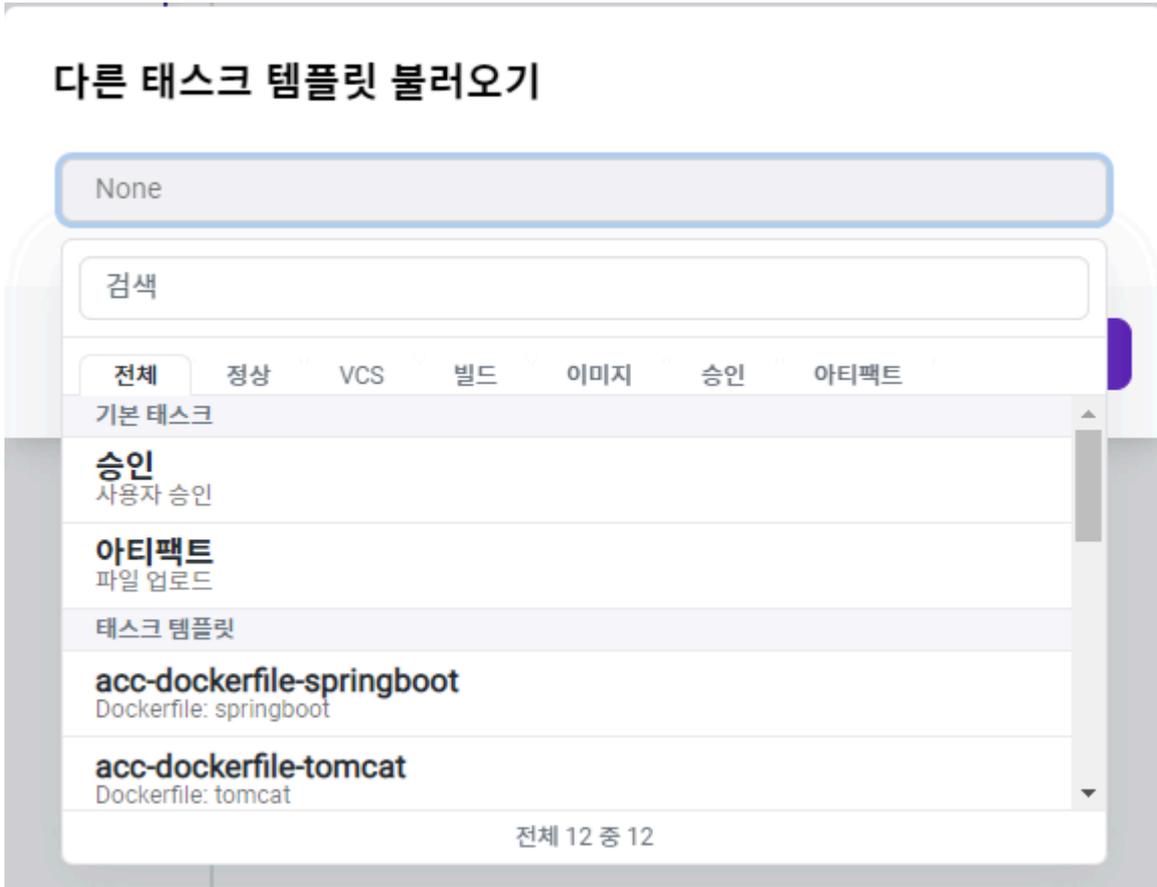
TIP



CAUTION

파이프라인의 작업공간에 대한 프로비저닝 설정 구성 시, 접근 모드를 ReadWriteOnce로 구성할 경우 위 tip의 예시와 같이 연관관계를 child 와 child-2 테스트가 같이 동시에 시작하는 파이프라인을 구성하려고 하는 경우, 쓰기가 실패할 수 있기 때문에 ReadWriteMany로 생성해야 한다.

태스크를 작성하기 위해서는 태스크 템플릿에서 필요한 태스크를 불러와 파라미터를 설정하는 과정을 거친다.



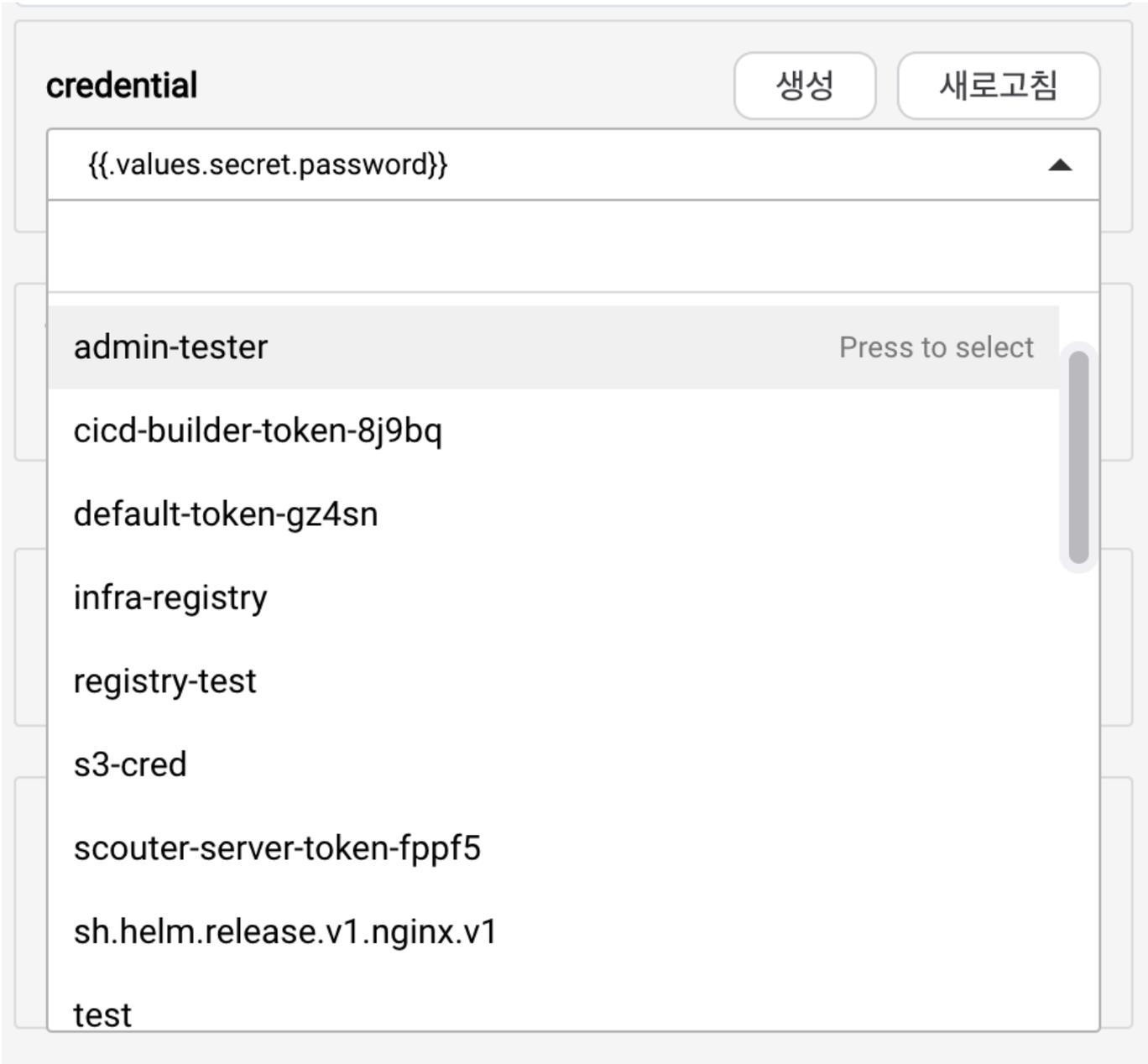
태스크 템플릿의 종류는 다음과 같다.

종류	설명
normal	일반 유형
vcs	Version Control System 유형 (e.g. git, svn), 만약 git 유형으로 AWS CodeCommit ssh 인증을 사용할 경우, kubernetes.io/ssh-auth 타입 시크릿에 data.ssh-login-name 에 ssh에 사용되는 login_name 값을 반드시 기술해야 한다.
build	소스빌드 유형 (e.g. maven, ant, gradle)
image	컨테이너 이미지를 만드는 유형 (e.g. kaniko)
승인	승인 태스크로 spec.tasks[].approvers 에 대해 반드시 기술하지만 container를 기술하지는 않음
아티팩트	파이프라인마다 설정되어 있는 S3저장소에 업로드한 데이터를 조회 spec.tasks[].buckets 을 반드시 기술하지만 container를 기술하지는 않음

파라미터까지 설정이 완료되면 저장 버튼을 선택해 작성을 완료한다.

NOTE | 파이프라인 수정 시 쿠버네티스 리소스를 참조하는 방법은 다음과 같다.

- 현재 사용자 권한에 따른 쿠버네티스 리소스 목록을 선택
- 환경변수를 입력



- 파이프라인 수정 시 쿠버네티스 리소스를 생성할 수 있는 경우는 Secret, ConfigMap, PersistentVolumeClaim 3가지 경우가 있다. 생성 화면은 각 메뉴별 생성 화면과 동일하다.

파이프라인 실행 미리보기

이력 탭에서 실행 버튼을 클릭하면 파이프라인 미리보기 페이지로 이동할 수 있다.

클러스터 jhchae-host-cluster 네임스페이스 junhyeong admin

빌드 > 파이프라인

test-p 기본 + 파이프라인 생성

junhyeong 1 month ago

PIPELINE test-p

상태 Active

요약 -

태스크 1

마지막 빌드 / 버전 Succeeded / 21

일반 파이프라인 이력 트리거 YAML

pipeline test-p 실행 삭제

상태	버전 ↓	실행자	시작 시간	종료 시간	메시지	동작
✓	21	accordion:user:admin	2024-12-02 13:53:16	2024-12-02 13:53:27		재실행 변경사항
✓	20	accordion:user:admin	2024-12-02 13:52:46	2024-12-02 13:52:56		재실행 변경사항
✓	19	accordion:user:admin	2024-12-02 11:29:02	2024-12-02 11:29:12		재실행 변경사항
✓	18	accordion:user:admin	2024-12-02 11:19:31	2024-12-02 11:19:41		재실행 변경사항
✓	17	accordion:user:admin	2024-12-02 11:18:22	2024-12-02 11:18:32		재실행 변경사항
✓	16	accordion:user:admin	2024-12-02 11:11:38	2024-12-02 11:11:48		재실행 변경사항
✓	15	accordion:user:admin	2024-12-02 11:11:06	2024-12-02 11:11:16		재실행 변경사항
✓	14	accordion:user:admin	2024-12-02 10:57:35	2024-12-02 10:57:45		재실행 변경사항

클러스터 jhchae-host-cluster 네임스페이스 junhyeong admin

빌드 > 파이프라인 > 파이프라인 실행 미리보기

신규 빌드 실행

요약

test-p

현재 버전 21 2024-12-02 13:53:16 미리보기 22 -

```

7 | template:
8 |   container:
9 |     args:
10 |       - |
11 |         | {{.values.cmd}}
12 |     command:
13 |       - sh
14 |       - '-c'
15 |     image: 10.60.160.51:5000/busybox
16 |     name: ''
17 |     resources: {}
18 |     valueschema: {}
19 |     properties:
20 |       cmd:
21 |         description: bash shell command
22 |         format: shell
23 |         type: string
24 |     type: object
25 |   templateRef:
26 |     clusterScope: false
27 |     name: acc-shell
28 |   values:
29 |     cmd: ''
30 | status:
31 |   phase: Succeeded
32 |   version: 21
33 |
7 | template:
8 |   container:
9 |     args:
10 |       - |
11 |         | {{.values.cmd}}
12 |     command:
13 |       - sh
14 |       - '-c'
15 |     image: 10.60.160.51:5000/busybox
16 |     name: ''
17 |     resources: {}
18 |     valueschema: {}
19 |     properties:
20 |       cmd:
21 |         description: bash shell command
22 |         format: shell
23 |         type: string
24 |     type: object
25 |   templateRef:
26 |     clusterScope: false
27 |     name: acc-shell
28 |   values:
29 |     cmd: ''
30 | status:
31 |   lastBuildPhase: Succeeded
32 |   lastVersion: 21
33 |

```

> 이벤트 (NORMAL 2 / WARNING 0)

성공한 가장 최신의 빌드를 현재 버전 으로서 비교 기준이 되며, 빌드를 통해 반영될 파이프라인 변경 사항을 미리보기 버전 으로서 비교가 가능하다.

우측 상단의 실행버튼으로 파이프라인을 실행시킨다.

파이프라인 실행과 중지

이력 탭에서 실행 버튼을 선택하면 파이프라인의 빌드가 실행되는 모습을 확인할 수 있다.

The screenshot shows the 'test-p' pipeline configuration page. On the left, a summary card indicates the pipeline is 'Active' with 1 task and is currently 'Running / 23'. The main table lists previous runs, with the most recent run (ID 23) in a 'Running...' state. A red box highlights the '중지' (Stop) button in the top right corner and the stop icon in the '동작' (Action) column for the running task.

상태	버전 ↓	실행자	시작 시간	종료 시간	메시지	동작
Running...	23	accordion:user:admin	2025-01-07 16:17:44	Running...		중지
Success	22	accordion:user:admin	2025-01-07 16:17:28	2025-01-07 16:17:38		재실행, 변경사항
Success	21	accordion:user:admin	2024-12-02 13:53:16	2024-12-02 13:53:27		재실행, 변경사항
Success	20	accordion:user:admin	2024-12-02 13:52:46	2024-12-02 13:52:56		재실행, 변경사항
Success	19	accordion:user:admin	2024-12-02 11:29:02	2024-12-02 11:29:12		재실행, 변경사항
Success	18	accordion:user:admin	2024-12-02 11:19:31	2024-12-02 11:19:41		재실행, 변경사항
Success	17	accordion:user:admin	2024-12-02 11:18:22	2024-12-02 11:18:32		재실행, 변경사항

상단의 중지 버튼 또는 우측의 멈춤 아이콘 버튼을 선택하면 진행 중인 파이프라인의 빌드를 중지시킬 수 있다.

파이프라인 재실행과 미리보기

빌드 내역 중 특정 빌드로 다시 배포하고 싶은 경우 해당 빌드 내역의 우측 버튼을 이용해 다시 배포도 가능하다. 이 경우 다시 실행하려는 빌드가 성공으로 완료되었을 때만 가능하다.

재실행 버튼을 눌러서 재실행에 대한 파이프라인 미리보기 페이지로 이동할 수 있다.

빌드 > 파이프라인 > 파이프라인 재실행 미리보기

The screenshot shows a side-by-side comparison of pipeline configurations. The left pane shows the '현재 버전' (Current version) 23, and the right pane shows the '재실행할 버전' (Version to be re-executed) 22. Both configurations are identical, showing a single task with a shell command. The status for version 23 is 'Succeeded' and for version 22 is also 'Succeeded'.

```

1 apiVersion: cicd.accordions.co.kr/v1beta1
2 kind: Pipeline
3 spec:
4   tasks:
5     - category: normal
6       name: taskc
7       template:
8         container:
9           args:
10            - |
11              | {{.values.cmd}}
12            command:
13              - sh
14              - '-c'
15            image: 10.60.160.51:5000/busybox
16            name: ''
17            resources: {}
18            valueschema:
19              properties:
20                cmd:
21                  description: bash shell command
22                  format: shell
23                  type: string
24                type: object
25            templateRef:
26              clusterScope: false
27              name: acc-shell
28            values:
29              cmd: ''
30 status:
31   phase: Succeeded
32 version: 23
33
1 apiVersion: cicd.accordions.co.kr/v1beta1
2 kind: Pipeline
3 spec:
4   tasks:
5     - category: normal
6       name: taskc
7       template:
8         container:
9           args:
10            - |
11              | {{.values.cmd}}
12            command:
13              - sh
14              - '-c'
15            image: 10.60.160.51:5000/busybox
16            name: ''
17            resources: {}
18            valueschema:
19              properties:
20                cmd:
21                  description: bash shell command
22                  format: shell
23                  type: string
24                type: object
25            templateRef:
26              clusterScope: false
27              name: acc-shell
28            values:
29              cmd: ''
30 status:
31   phase: Succeeded
32 version: 22
33
  
```

성공한 가장 최신의 빌드를 현재 버전 으로서 비교 기준이 되며, 재실행을 통해 반영될 파이프라인 변경 사항을 재실행할 버전 으로서 비교가 가능하다.

우측 상단의 재실행버튼으로 파이프라인을 재실행시킨다.

파이프라인 버전별 비교하기

성공한 특정 빌드 버전들에 한하여 서로 변경 사항을 비교가능하다.

클러스터 jhchae-host-cluster 네임스페이스 junhyeong admin

빌드 > 파이프라인 > 파이프라인 이력 변경사항

test-p

버전 22 2025-01-07 16:17:28 비교 버전 23 2025-01-07 16:17:44

<pre> 1 apiVersion: ckd.accordions.co.kr/v1beta1 2 kind: Pipeline 3 spec: 4 tasks: 5 - category: normal 6 name: taskc 7 template: 8 container: 9 args: 10 - 11 {{.values.cmd}} 12 command: 13 - sh 14 - '-c' 15 image: 10.60.160.51:5000/busybox 16 name: '' 17 resources: {} 18 valueschema: 19 properties: 20 cmd: 21 description: bash shell command 22 format: shell 23 type: string 24 type: object 25 templateRef: 26 clusterScope: false 27 name: acc-shell 28 values: 29 cmd: '' 30 status: 31 phase: Succeeded 32- version: 22 33 </pre>	<pre> 1 apiVersion: ckd.accordions.co.kr/v1beta1 2 kind: Pipeline 3 spec: 4 tasks: 5 - category: normal 6 name: taskc 7 template: 8 container: 9 args: 10 - 11 {{.values.cmd}} 12 command: 13 - sh 14 - '-c' 15 image: 10.60.160.51:5000/busybox 16 name: '' 17 resources: {} 18 valueschema: 19 properties: 20 cmd: 21 description: bash shell command 22 format: shell 23 type: string 24 type: object 25 templateRef: 26 clusterScope: false 27 name: acc-shell 28 values: 29 cmd: '' 30 status: 31 phase: Succeeded 32+ version: 23 33 </pre>
--	--

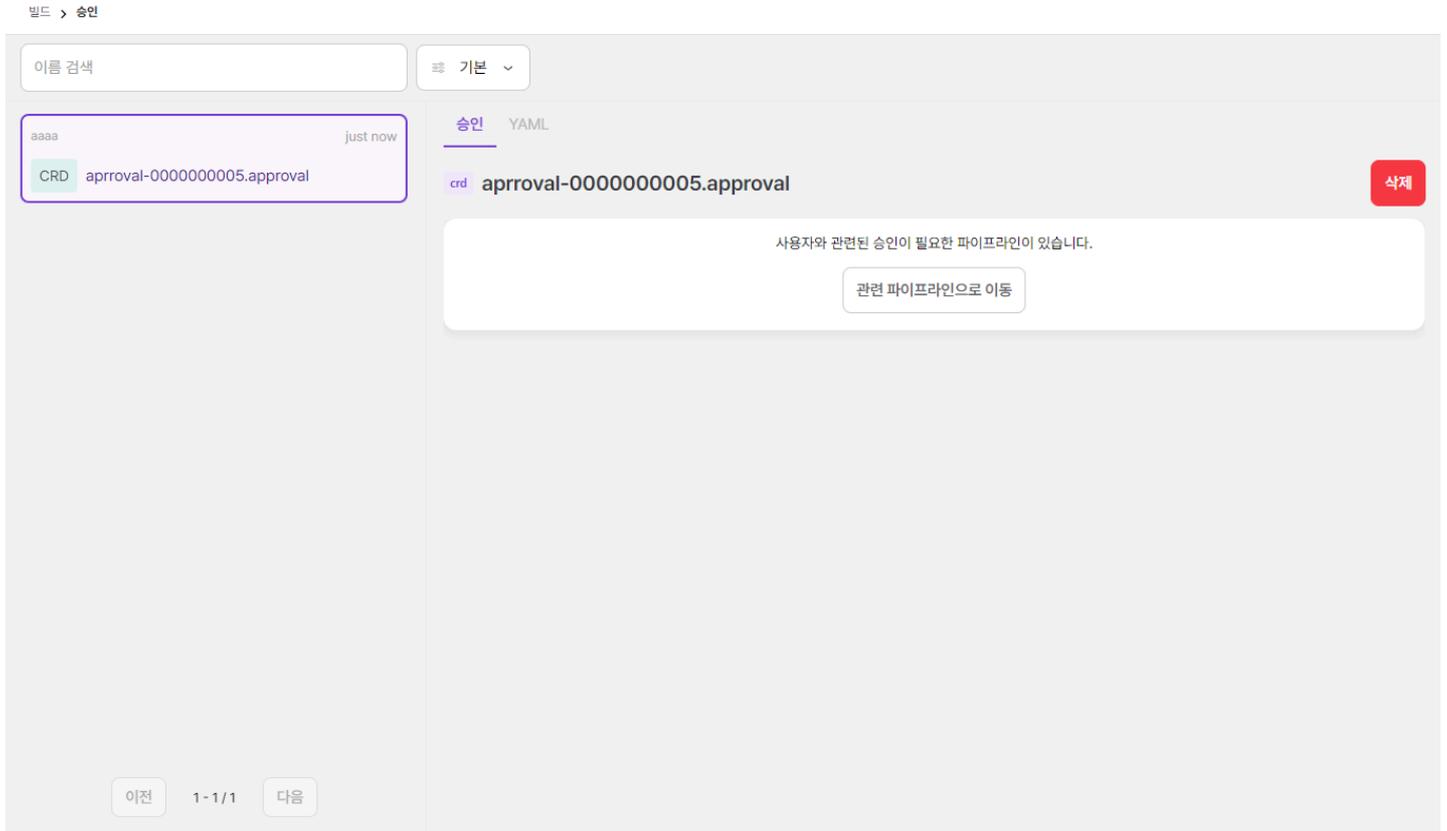
리스트에서 선택한 특정 버전을 기준으로 다른 성공한 버전들과 비교할 수 있다.

트리거

카탈로그의 트리거와 사용방법이 동일하기 때문에 해당 가이드를 참고 한다.

4.3.3.2. 승인

파이프라인의 빌드를 실행 중 승인이 필요한 태스크가 있으면 승인자의 승인이 완료될 때까지 대기한다. 대기시간은 기본 30분으로 설정되어 있으며 대기시간 내 승인이 이루어지지 않는 경우 미승인으로 처리된다. 승인은 사용자가 생성하거나 삭제하는 것이 아니라 파이프라인에 의해 관리된다.



4.3.3.3. 태스크 템플릿

파이프라인 생성/수정 시 태스크 작성을 위해 필요한 태스크 템플릿을 관리한다. Preview 탭에서는 사용자에게 제공되는 폼을 미리 확인할 수 있다.

빌드 > 태스크 템플릿

이름 검색

기본

+ 태스크 템플릿 생성

aaaa 3 months ago

TEMPLATE acc-copy-to-pvc

요약 copy: to pvc

aaaa 4 months ago

TEMPLATE acc-dockerfile-springboot

요약 Dockerfile: springboot

aaaa 4 months ago

TEMPLATE acc-dockerfile-tomcat

요약 Dockerfile: tomcat

aaaa 4 months ago

TEMPLATE acc-dockerfile-wildfly

요약 Dockerfile: wildfly

aaaa 4 months ago

이전 1 - 17 / 17 다음

Preview YAML

template acc-copy-to-pvc

삭제

pvc ?

생성 수정

uploadFile ?

4.3.4. 워크로드

네임스페이스 스코프의 워크로드 메뉴는 클러스터 스코프의 워크로드 메뉴와 사용 방법이 동일하기 때문에 해당 가이드를 참고한다.

4.3.5. 구성

네임스페이스 스코프의 구성 메뉴는 클러스터 스코프의 구성 메뉴와 사용 방법이 동일하기 때문에 해당 가이드를 참고한다.

4.3.6. 네트워크

네임스페이스 스코프의 네트워크 메뉴는 클러스터 스코프의 네트워크 메뉴와 사용 방법이 동일하기 때문에 해당 가이드를 참고한다.

4.3.7. 스토리지

네임스페이스 스코프의 스토리지 메뉴는 클러스터 스코프의 스토리지 메뉴의 퍼시스턴트볼륨클레임과 사용 방법이 동일하기 때문에 해당 가이드를 참고한다.

4.3.8. 접근제어

네임스페이스 스코프의 접근제어 메뉴는 클러스터 스코프의 접근제어 메뉴의 룰, 롤바인딩, 서비스어카운트와 사용 방법이 동일하기 때문에 해당 가이드를 참고한다.

4.3.9. 모니터링

모니터링은 운영자가 안정적인 시스템 운영을 할 수 있도록 시스템 리소스와 애플리케이션 퍼포먼스 및 로그 (이벤트 로그, 감사 로그, 알림 로그) 에 대한 정보를 제공한다.

NOTE

시스템, 이벤트 로그, 감사 로그, 알림 로그 메뉴의 경우 클러스터 스코프의 모니터링 메뉴에도 동일한 메뉴가 있다. 둘은 스코프의 기준이 클러스터와 네임스페이스라는 것을 제외하고 동일하기 때문에 해당 가이드를 참고한다.

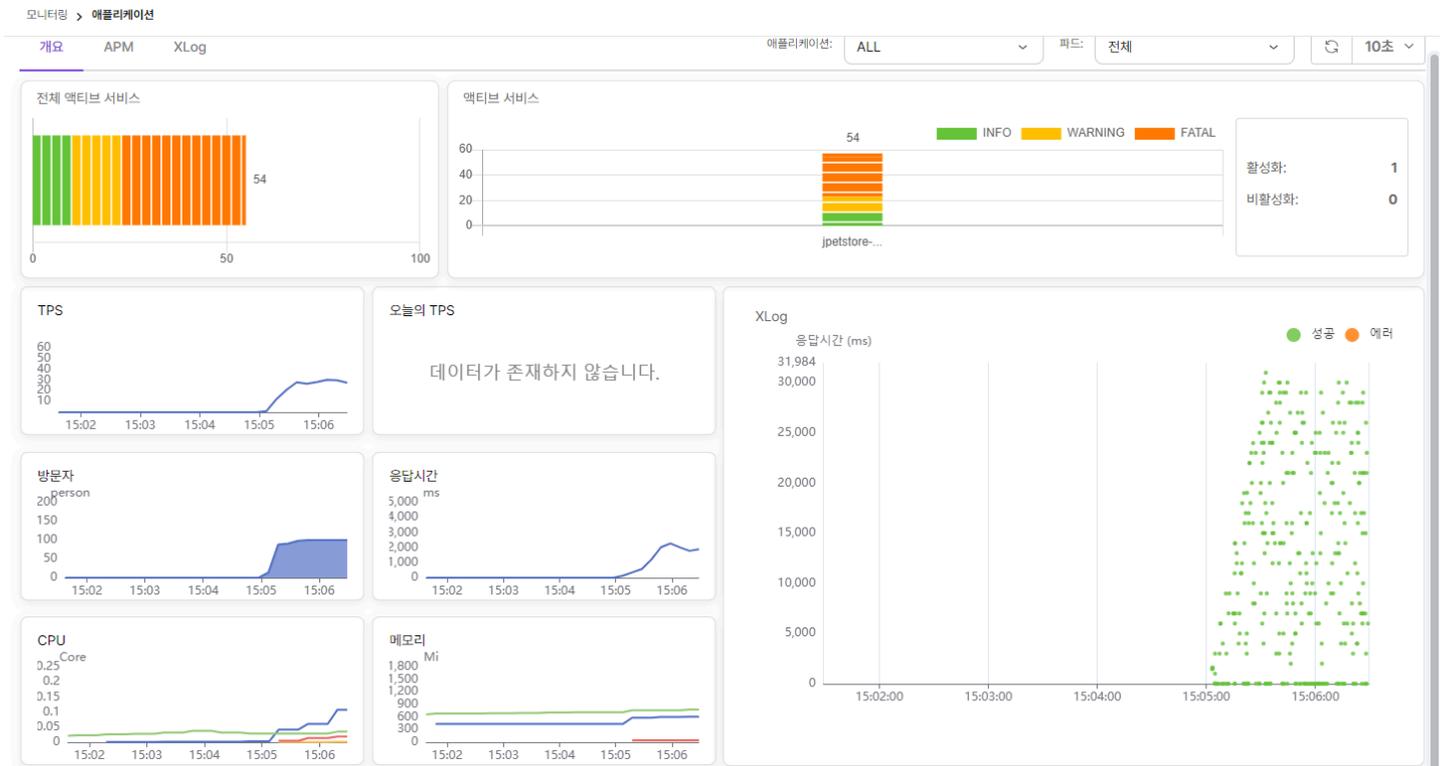
- 시스템: 클러스터/시스템
- 이벤트 로그: 클러스터/이벤트 로그
- 감사 로그: 클러스터/감사 로그
- 알림 로그: 클러스터/알림 로그

4.3.9.1. 애플리케이션

개요

애플리케이션 개요에서는 수집한 메트릭을 중심으로 애플리케이션의 성능 모니터링 정보를 제공한다. 주요 모니터링 항목은 다음과 같다.

항목	메트릭
서비스	전체 액티브 서비스, 액티브 서비스, TPS, 오늘의 TPS, 방문자, 응답시간, XLog
시스템	CPU, 메모리



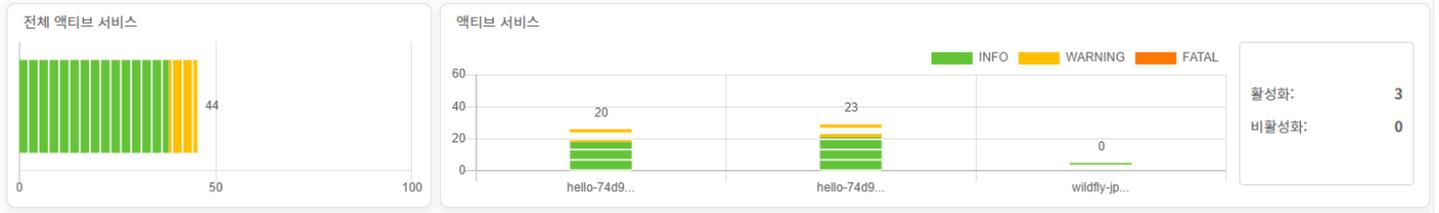
각 모니터링 항목에 대한 설명은 다음과 같다.

액티브 서비스

개별 인스턴스 별로 현재 처리중인 리퀘스트에 대한 정보를 바 차트로 제공한다. 처리중인 리퀘스트는 단계별로 만족(Info), 허용(Warning), 불만(Fatal) 으로 구분하고 각각을 녹색, 노란색(3초), 주황색(8초)으로 표시한다.

- 전체 액티브 서비스는 현재 처리중인 전체 인스턴스의 상태 정보를 제공한다.
- 액티브 서비스는 현재 처리중인 개별 인스턴스 상태 정보를 제공한다.

처리중인 리퀘스트들이 표시될 때, 해당 차트 선택하면 액티브 서비스 정보 모달이 나타나고 진행중인 리퀘스트에 대한 정보를 얻을 수 있다.



액티브 서비스 정보

액티브 서비스 에서 차트를 선택하면 나타나는 모달로 정보를 제공한다. 쓰레드에 대한 메타데이터와 진행중인 스택 정보를 볼 수 있다. 쓰레드가 지연되고 있으면 스택에서 이슈가 있을 확률이 높아 해당 스택을 분석할 필요가 있다.

NOTE

액티브 서비스 정보

#	파드명	응답시간(ms)	서비스	IP	TXID	상태	쓰레드 CPU 시간	노트
1	jpetstore-7c89f8bf9d-szq49	0	/actions/Sleep.action	10.20.200.225		end	0	-
2	jpetstore-7c89f8bf9d-szq49	28,844	/actions/Sleep.action	10.20.200.225	x449t745g2mkfp	TIMED_WAITING	722	-
3	jpetstore-7c89f8bf9d-szq49	26,882	/actions/Sleep.action	10.20.200.225	x7gagb92fk0n44	TIMED_WAITING	1,365	-
4	jpetstore-7c89f8bf9d-szq49	25,687	/actions/Sleep.action	10.20.200.225	x65ujca77a3j	TIMED_WAITING	584	-

액티브 서비스 상세정보

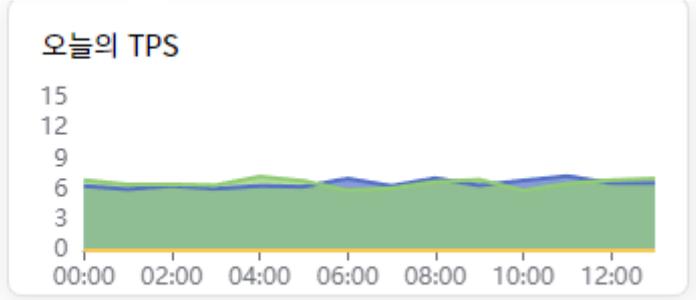
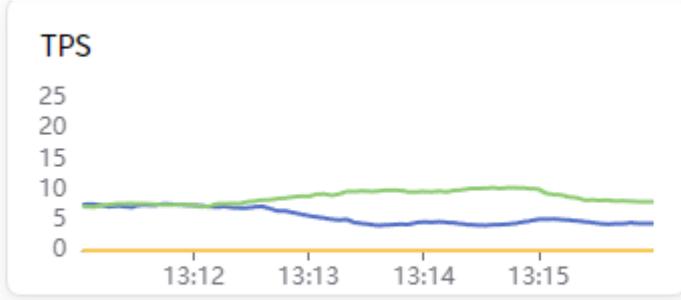
파드명	jpetstore-7c89f8bf9d-szq49	쓰레드 CPU 시간	42,697 ms
차단된 개수	49	쓰레드 ID	461
장군명	-	쓰레드명	http-nio-8080-exec-433
응답시간	5,852 ms	쓰레드 사용자 시간	24,600 ms
상태	TIMED_WAITING	대기 횟수	54,909
		대기 시간	-1 ms

```

프로필
java.lang.Thread.sleep(Native Method)
org.apache.jsp.sleep_jsp._jspService(sleep_jsp.java:118)
org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
  
```

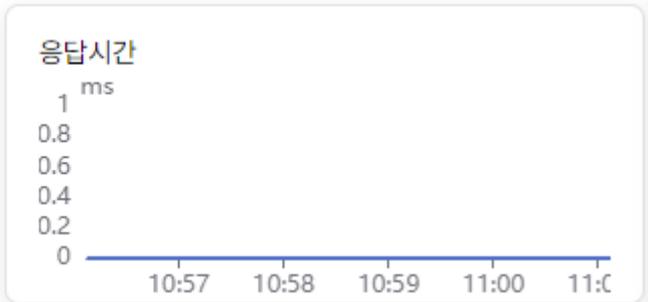
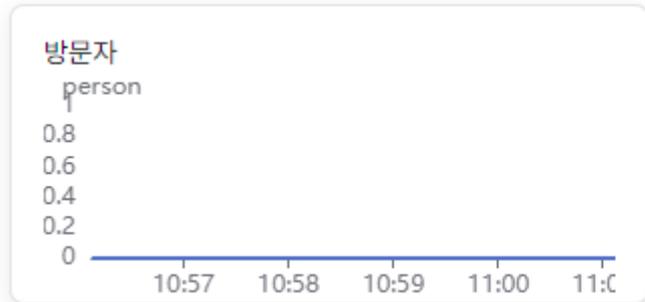
TPS, 오늘의 TPS

TPS는 초당 처리량을 의미하고 오늘의 TPS는 당일 시간대별 TPS의 평균을 의미한다.



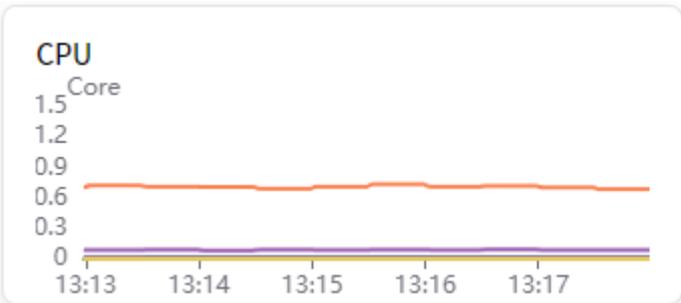
방문자, 응답시간

방문자는 5분간 방문한 유니크한 방문자 수를, 응답시간은 애플리케이션 응답시간을 의미한다.



CPU, 메모리

CPU와 메모리 사용량을 제공한다.

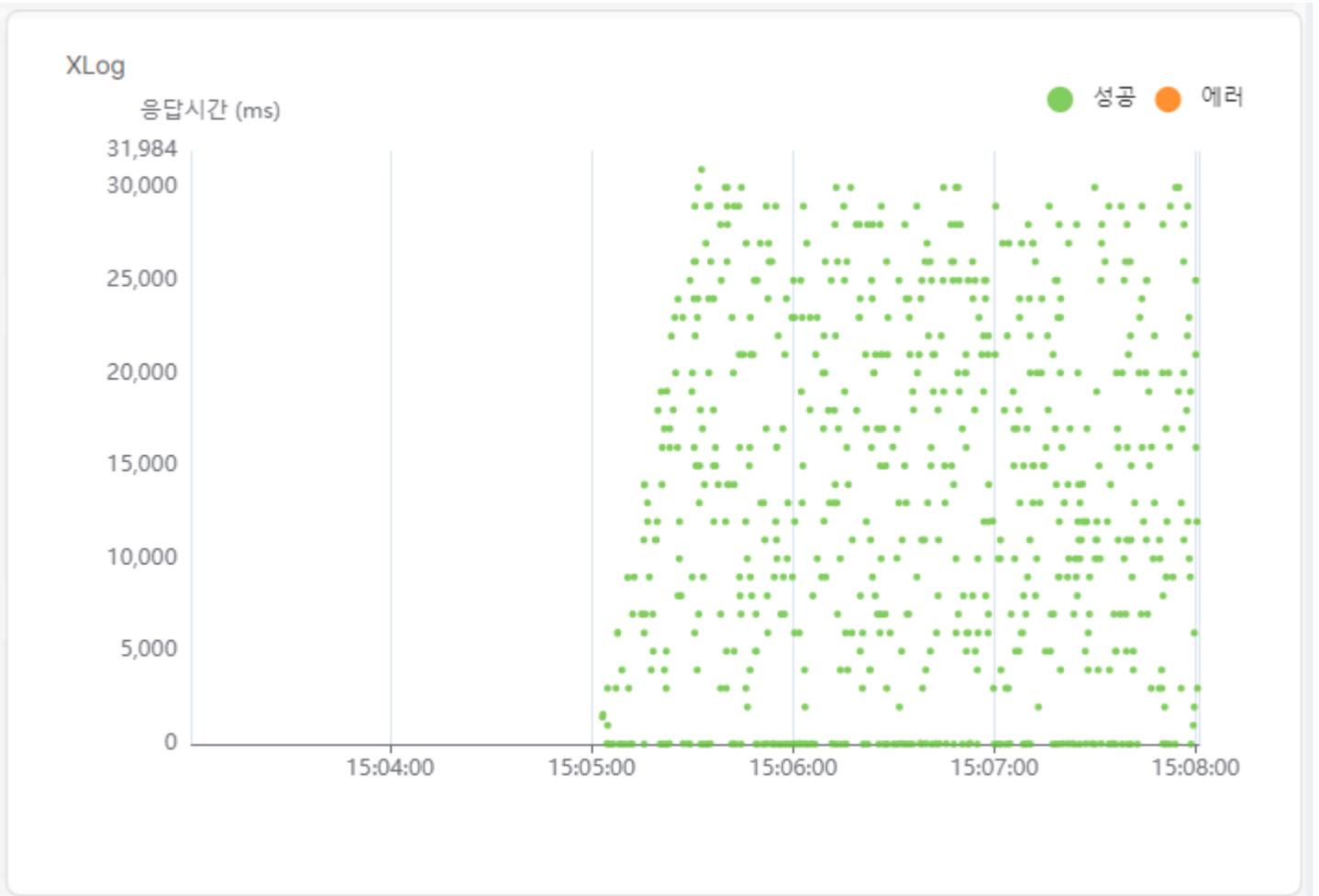


XLog

시간별 응답시간 정보를 scatter 산점도 차트로 표현한다. 차트에서 X 축은 요청이 발생한 시간을 Y 축은 응답시간을 의미한다. 해당 요청에 대한 HTTP 응답 코드에 따라 차트에 표시되는 색이 녹색 또는 주황색으로 표시되며 주황색은 오류 발생(HTTP 5xx 코드)을 의미한다.

차트의 일부 영역을 마우스로 블록 지정하면 XLog Info에서 해당 영역에서 처리된 요청들에 대한 상세 프로파일을 분석할 수 있다.

NOTE 콘솔에서 표시되는 XLog 최대 응답 크기의 기본값은 20MB이다.



XLog Info 에서는 XLog 산점도 차트에서 선택한 영역을 기준으로 실행된 요청에 대한 정보를 나열한다. 특정 요청을 선택하면 해당 요청에 대한 시작시간, 수행시간, CPU 소요 시간 등의 정보를 확인할 수 있다.

XLog 드래그 최대 개수는 기본 2000개로 설정되어 있으며, 이를 초과하면 드래그 범위를 줄이거나 설정을 변경해야 한다.

NOTE

성공	2000	실패	0	전체	2000	!
----	------	----	---	----	------	---

XLog 드래그 최대 개수 초과

상세 XLog SQL 파라미터 바인드

애플리케이션명	jpetstore	응답시간	18003 ms
파드명	jpetstore-7c89f8bf9d-szq49	시작 시간	2024-09-03 15:08:59
TXID	x6ov8b09v02550	종료 시간	2024-09-03 15:09:17
서비스	/actions/Sleep.action	API 개수	0
IP 주소	10.20.200.225	API 시간	0 ms
CPU	0 ms	SQL 개수	0
메모리	0 kbytes	SQL 시간	0 ms
유저 에이전트	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.100 Safari/537.36		
에러			

각 항목에 대한 내용은 다음과 같다.

항목	설명
애플리케이션명	애플리케이션 이름
파드명	파드 이름
Txid	트랜잭션 ID
서비스	서비스 URL
IP 주소	요청이 발생한 IP
Cpu	서비스 CpuTime
메모리	서비스에서 사용한 memory
응답시간	서비스 소요시간
시작 시간	서비스 시작시간
종료 시간	서비스 종료시간
API 개수	서비스에서 API 수행 횟수
API 시간	서비스에서 사용된 API 시간의 합
SQL 개수	서비스에서 SQL 수행 횟수
SQL 시간	서비스에서 사용된 SQL 시간의 합

프로파일에서는 해당 메소드의 실행에 소요된 시간을 표시한다. 메소드들의 실행 관계는 트리 형태로 표현하며 이전 단계와 시간 차이를 T-GAP 으로 표시한다.

p#	#	TIME	T-GAP	CPU	CONTENTS
	[*****]	00:48:01.257	0	0	start transaction
-	[000000]	00:48:01.257	0	0	[driving thread] http-nio-8080-exec-10
-	[000001]	00:48:01.258	1	0	param: ko_KR
-	[000002]	00:48:01.258	0	0	param: {}
-	[000003]	00:48:01.258	0	0	param: org.apache.catalina.connector.ResponseFacade@7eafe61b
-	[000004]	00:48:01.258	0	0	param: excute
-	[000005]	00:48:01.258	0	0	param: H2DS
-	[000007]	00:48:01.260	1	0	PRE> SELECT l_returnflag, l_linestatus, SUM(l_quantity) AS sum_qty, SUM(l_extendedprice) AS sum_base_price, SUM(l_extendedprice * (@{1} - l_discount)) AS sum_disc_price, SUM(l_extendedprice * (@{2} - l_discount) * (@{3} + l_tax)) AS sum_charge, AVG(l_quantity) AS avg_qty, AVG(l_extendedprice) AS avg_price, AVG(l_discount) AS avg_disc, COUNT(*) AS count_order FROM lineitem WHERE l_shipdate <= DATE '@{4}' - INTERVAL '@{5}' DAY GROUP BY l_returnflag, l_linestatus ORDER BY l_returnflag, l_linestatus [1,1,1,'1998-12-01','108'] 1 ms

XLog 에서는 응답시간에 따라 일정 확률로 로그를 남긴다.

응답시간	확률
0~1초 사이	3%
1~2초 사이	5%
2~3초 사이	20%
3초 이상	100%

XLog에서 Bind SQL Parameter 를 활성화하면 프로파일에서 SQL문에 바인딩 된 실제 값을 표시할 수 있다.

```

-----
p#      #      TIME      T-GAP  CPU      CONTENTS
-----
- [*****] 00:48:01.257    0    0 start transaction
- [000000] 00:48:01.257    0    0 [driving thread] http-nio-8080-exec-10
- [000001] 00:48:01.258    1    0 param: ko_KR
- [000002] 00:48:01.258    0    0 param: {}
- [000003] 00:48:01.258    0    0 param: org.apache.catalina.connector.ResponseFacade@7eafe61b
- [000004] 00:48:01.258    0    0 param: excute
- [000005] 00:48:01.258    0    0 param: H2DS
- [000007] 00:48:01.260    1    0 PRE> SELECT l_returnflag, l_linestatus,
SUM(l_quantity) AS sum_qty,
SUM(l_extendedprice) AS sum_base_price,
SUM(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
SUM(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
AVG(l_quantity) AS avg_qty,
AVG(l_extendedprice) AS avg_price,
AVG(l_discount) AS avg_disc,
COUNT(*) AS count_order
FROM lineitem
WHERE l_shipdate <= DATE '1998-12-01' - INTERVAL '108' DAY GROUP BY l_returnflag, l_linestatus
1 ms

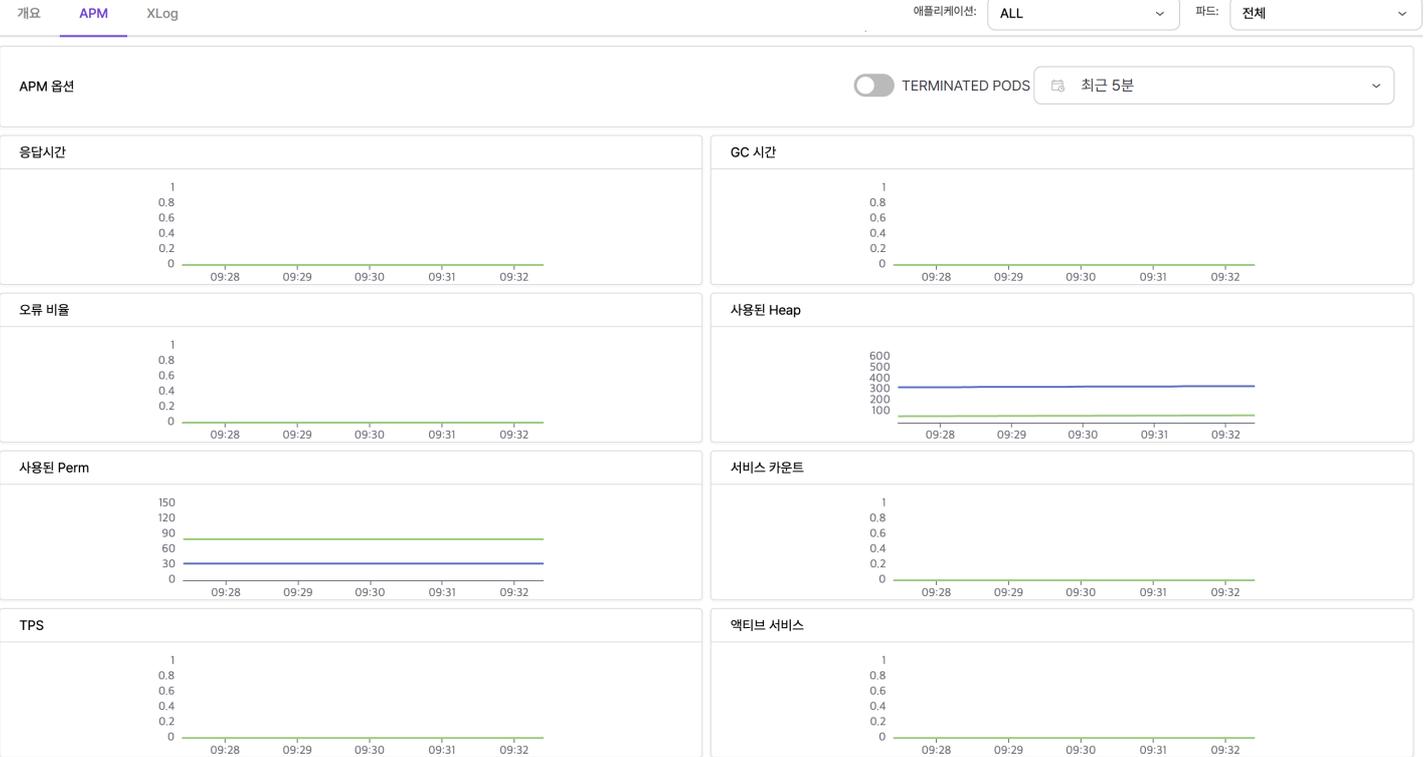
```

APM

APM 탭에서는 수집한 메트릭을 개요보다 더 상세하게 기간별로 제공한다. 주요 모니터링 항목은 다음과 같다.

항목	메트릭
서비스	응답 시간, GC 시간, 오류 비율, 사용된 Heap, 사용된 Perm, 서비스 카운트, TPS, 액티브 서비스

모니터링 > 애플리케이션



개요에서 설명을 한 항목을 제외하고 추가된 항목에 대한 설명은 다음과 같다.

GC 시간

GC 시간은 Garbage Collection 시간을 의미하며, GC 시간이 길어지면 서비스 응답 시간이 느려질 수 있다.

오류 비율

오류 비율은 서비스에서 발생한 오류 비율을 의미한다.

사용된 Heap, 사용된 Perm

사용된 Heap 은 Heap Memory 사용량을, 사용된 Perm 은 Perm Memory 사용량을 의미한다.

서비스 카운트

서비스 카운트는 서비스 호출 횟수를 의미한다.

XLog

XLog 탭에서는 수집한 XLog를 개요보다 더 상세하게 기간별로 제공한다. 기간별 조회를 제외하고 이전 설명과 기능은 동일하다.

4.3.10. 설정

네임스페이스 스코프의 설정 메뉴에서는 사용자에게 네임스페이스에 대한 권한을 할당하거나 알림 정책 및 수신자를 설정할 수 있다. 본 장에서는 알림 정책 메뉴에서 네임스페이스에 배포된 워크로드에 대한 알림 정책을 설정하는 방법에 대해 설명한다.

NOTE

- 네임스페이스 멤버 메뉴의 경우 클러스터 스코프의 클러스터 멤버 메뉴와 설정 방법이 동일하기 때문에 해당 가이드를 참고한다.

4.3.10.1. 네임스페이스 수신자

네임스페이스 스코프의 네임스페이스 수신자 메뉴는 글로벌 스코프의 글로벌 수신자 메뉴와 사용 방법이 동일하기 때문에 해당 가이드를 참고한다.

4.3.10.2. 알림 정책

클러스터 스코프의 알림 정책과 네임스페이스 스코프의 알림 정책의 차이점은 아래와 같다.

- 대상 지정 에서 노드 선택이 제외된다.
- Query 에서 Prometheus Query 작성 시 해당 네임스페이스의 Vector만 조회된다.

Vector에 해당 네임스페이스의 vector selector가 모두 존재하여야 한다.
라벨 키는 "namespace" 명칭이 포함되어야 한다.

예시

sample 네임스페이스의 경우 위의 규칙에 따라 아래와 같은 Query 입력이 가능하다.

- `APM_TPS{exported_namespace="sample"}`
- `kube_pod_info{namespace="sample"}`
- `apm_TPS{exported_namespace="sample"} * on(pod) group_left(created_by_kind) kube_pod_info{namespace="sample"}`

그 외에 클러스터 스코프의 알림 정책과 사용 방법이 동일하기 때문에 해당 가이드를 참고한다.