# 아코디언 사용자 설명서

Version 2.9

## 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. 인그레스
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. 롤
4.2.11.2. 롤바인딩
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

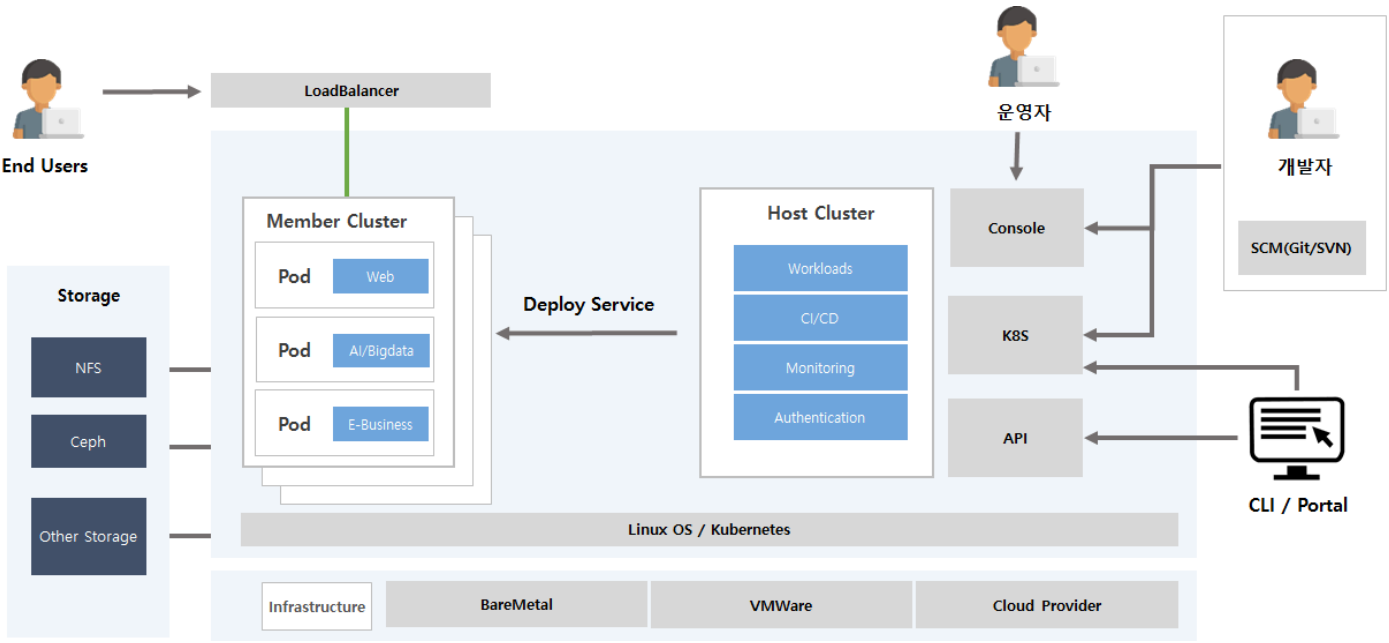
아코디언은 컨테이너 기반의 애플리케이션을 손쉽게 배포하고 자동화된 운영관리를 제공하기 위한 올인원 솔루션이다.

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.

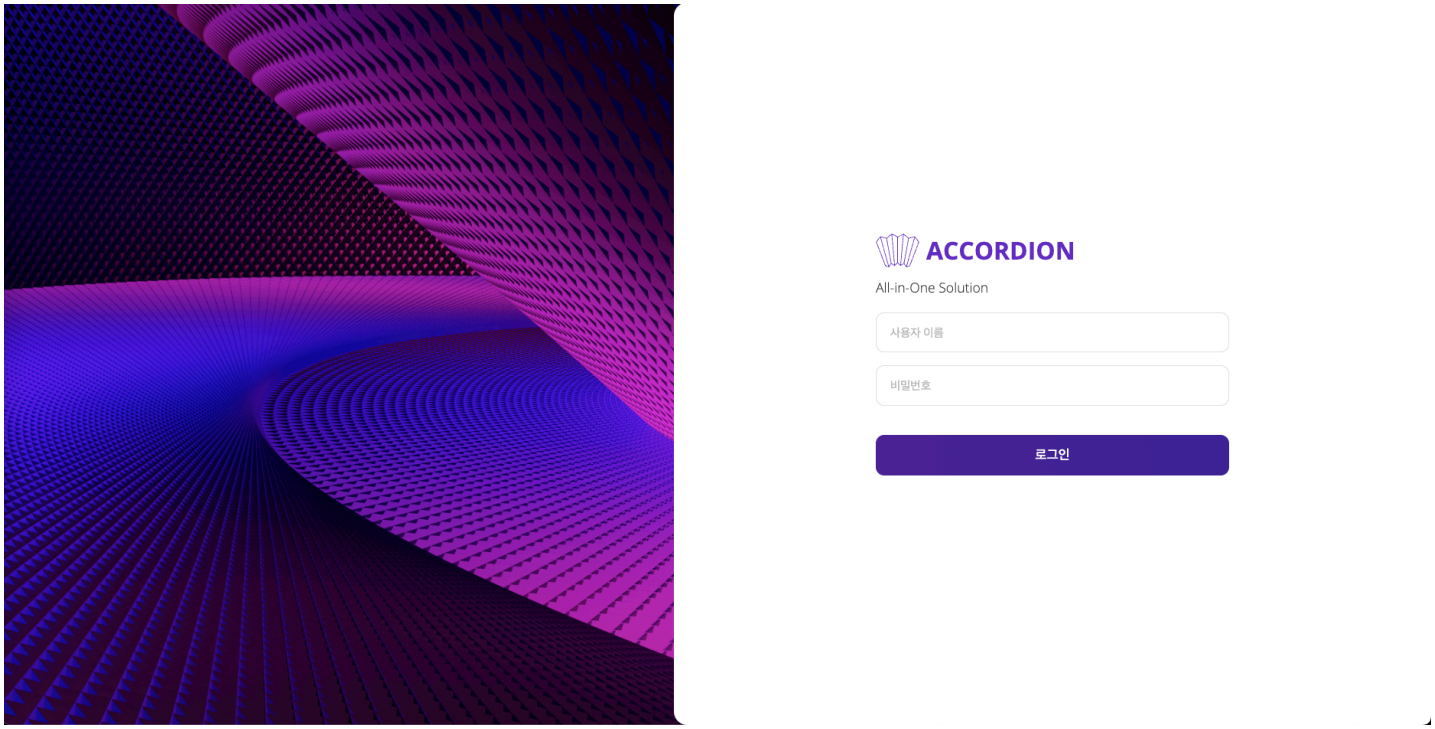
> *자세한 설명은 글로벌 권한/클러스터 권한/네임스페이스 권한을 참고한다.*

# 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.




## 2.2. 계정

초기 계정 정보는 설치 매뉴얼을 참고한다.

# 3. 화면 구성

## 3.1. 화면 영역

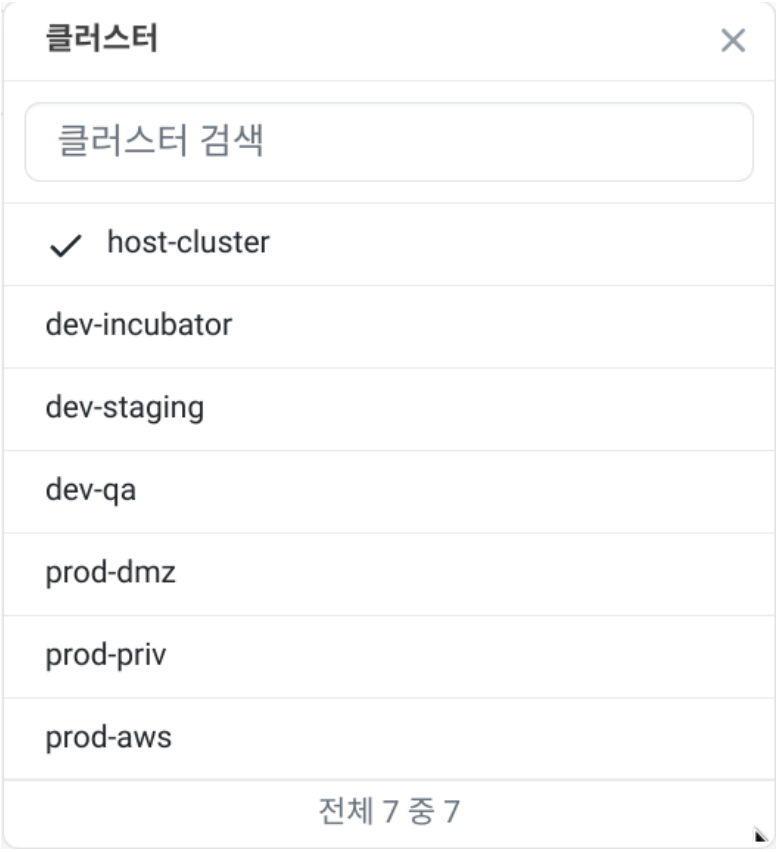
화면은 다음과 같이 3개의 영역으로 나누어진다.

| 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

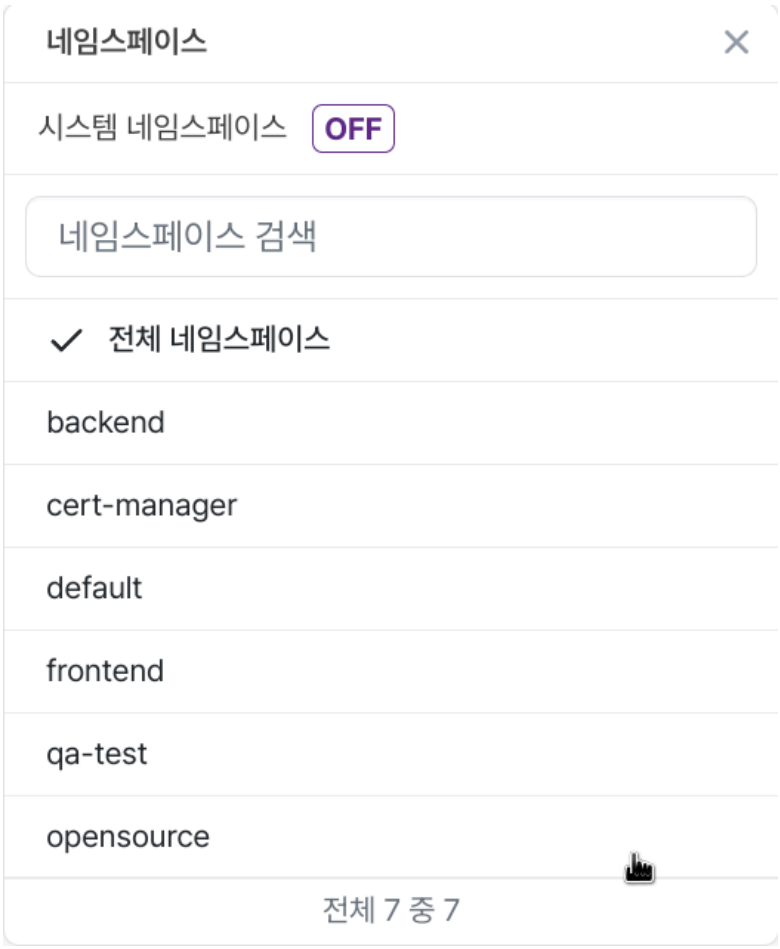
#### 3.1.1.1. 스코프 선택

클러스터 선택




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

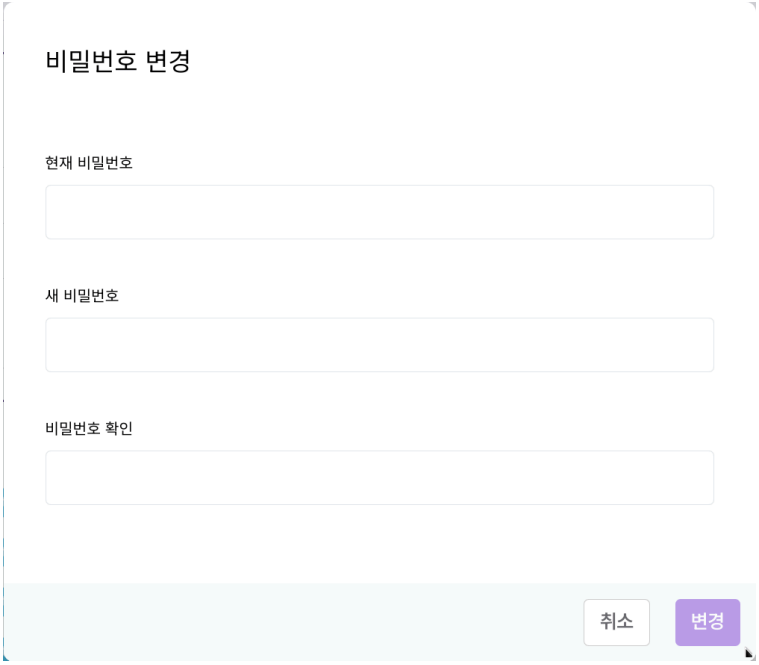
네임스페이스 선택




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

## 3.1.1.2. 사용자 설정

### 비밀번호 변경

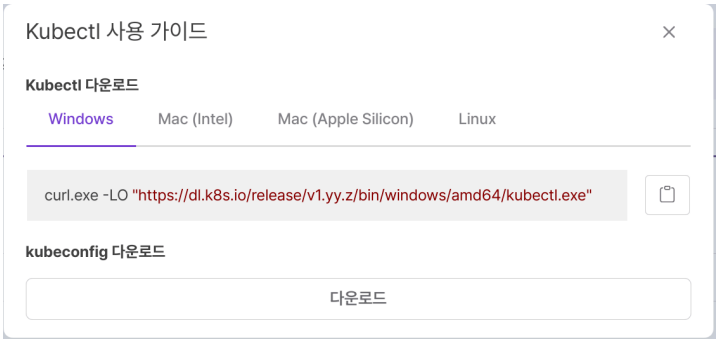



사용자의 비밀번호를 변경할 수 있다.

### 매뉴얼

아코디언 매뉴얼을 확인할 수 있다.

### kubectl 사용 가이드

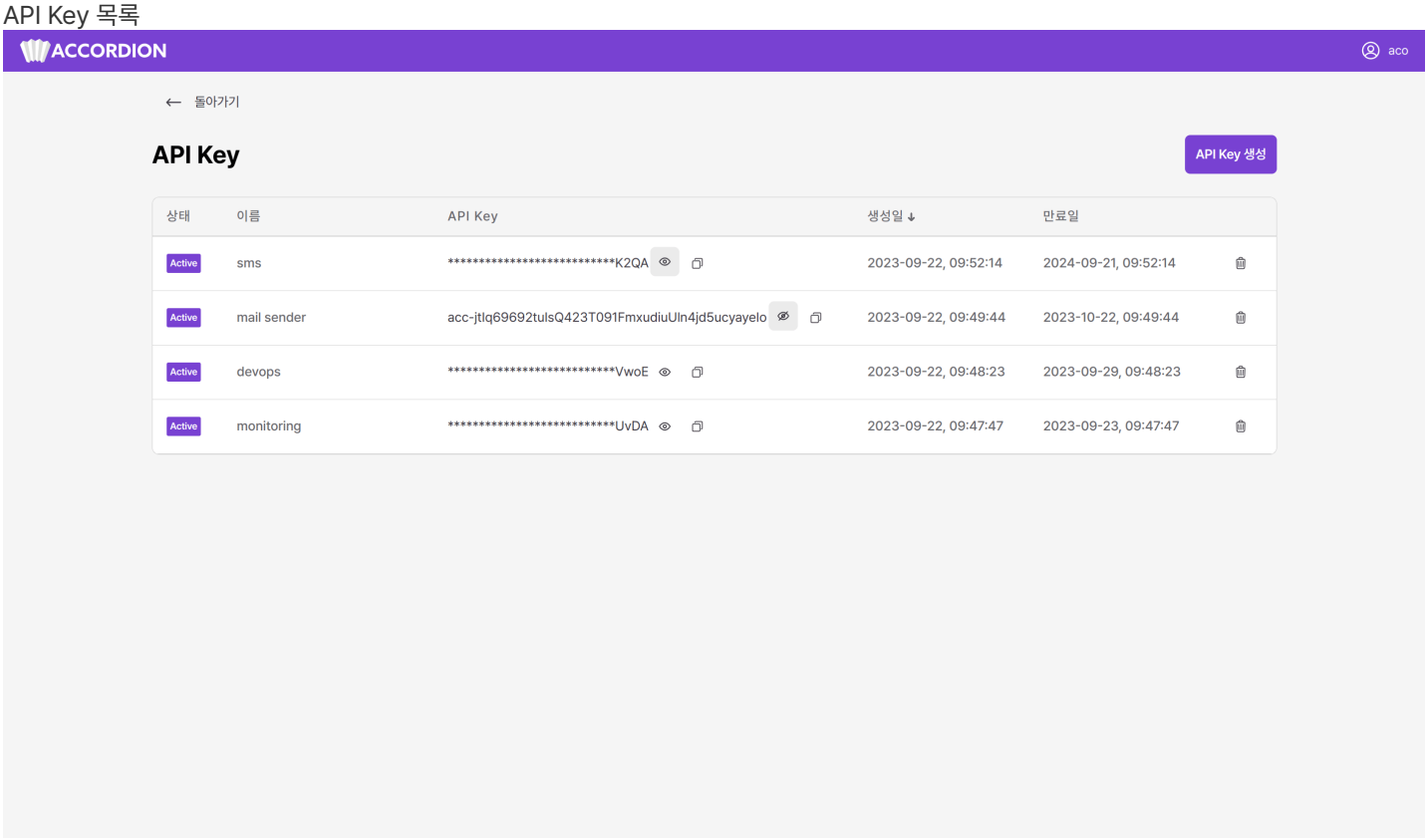



- 아코디언에 버전에 맞는 `kubectl` 실행 파일 다운로드 경로를 알 수 있다.
- 아코디언 사용자의 `kubeconfig` 를 다운로드 받을 수 있다.

## API Key

아코디언 API를 이용할 수 있는 token 을 관리한다.

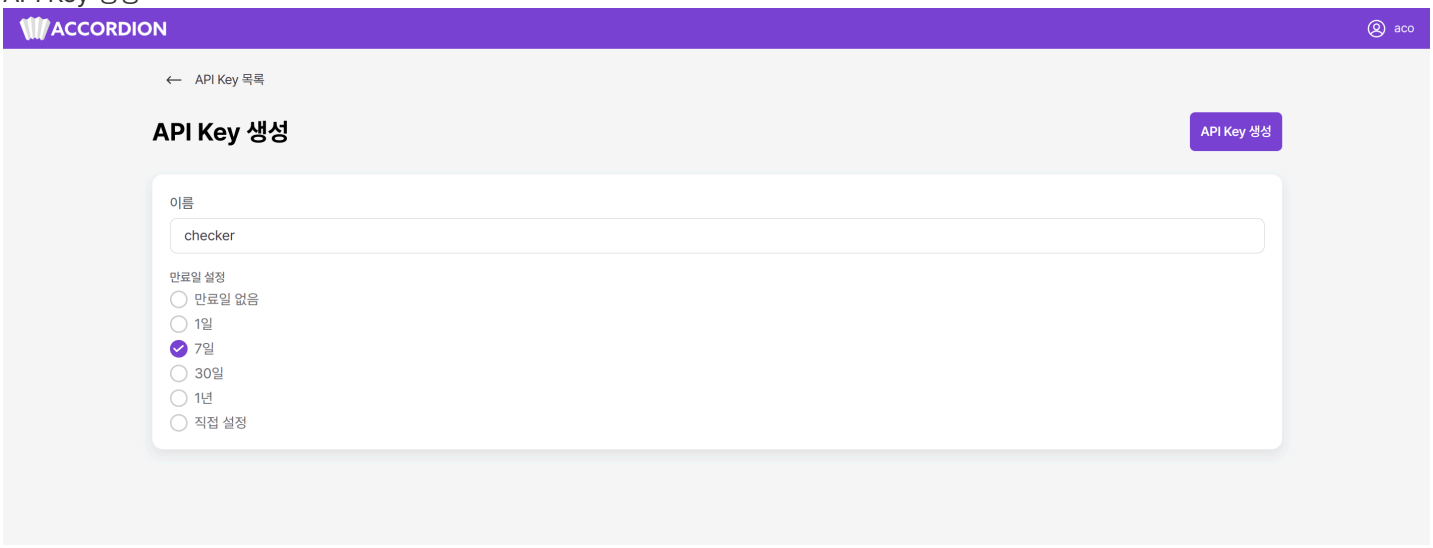
### API Key 목록



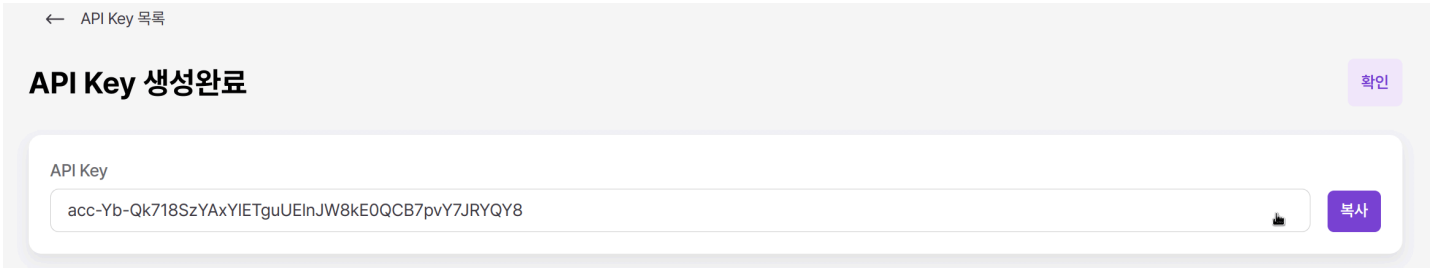

- 생성한 token 정보를 조회한다.

| 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 | 아코디언 전반에 걸쳐 계정 정보를 관리한다.<br>사용자, 그룹 및 그리고 글로벌/클러스터/네임스페이스 스코프 별 메뉴에 대한 권한을 관리하고 사용자/그룹과 글로벌 권한을 바인딩하여 글로벌 멤버로 설정한다.<br>그 외에 사용자의 접속 로그를 조회할 수 있다.<br>• 사용자: 사용자 정보를 조회, 관리<br>• 그룹: 사용자 그룹 정보를 조회, 관리<br>• 글로벌 권한: 글로벌 스코프의 메뉴에 대해 사용 권한을 집합하여 단일 권한으로 관리<br>• 클러스터 권한: 클러스터 스코프의 메뉴에 대해 사용 권한을 집합하여 단일 권한으로 관리<br>• 네임스페이스 권한: 네임스페이스 스코프의 메뉴에 대해 사용 권한을 집합하여 단일 권한으로 관리<br>• 글로벌 멤버: 사용자/그룹과 글로벌 권한을 바인딩한다.<br>• 사용자 접속 로그: 사용자의 접속 로그를 조회한다. |
| 글로벌 설정 | 활성화 키와 알림 발송을 위한 글로벌 수신자를 관리한다.<br>• 활성화 키: 활성화 키 정보를 조회, 등록한다.<br>• 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.<br>• 워크로드 대시보드: 배포한 파드 정보를 차트로 표시한다.<br>• 파드: 파드 정보를 제공하고 관리한다. 컨테이너 로그 및 터미널을 이용해 접속할 수 있다.<br>• 디플로이먼트/스테이트풀셋: 파드를 관리하는 리소스로 편집기로 관리하며 오토스케일을 설정할 수 있다.<br>• 데몬셋: 노드마다 파드를 실행하도록 제어한다.<br>• 잡/크론잡: 특정 작업을 일시적/주기적으로 실행할 수 있도록 관리한다. |
| 구성 | 시스템과 애플리케이션 구성 및 설정에 필요한 정보를 관리한다.<br>• 컨피그맵/시크릿: 일반 정보 또는 기밀성 정보를 저장한다.<br>• HPA/리밋레인지: 수평적 오토스케일 및 리소스 할당을 제한한다. |
| 네트워크 | 애플리케이션 연동을 위해 네트워크 구성에 필요한 리소스를 관리한다.<br>• 서비스: 애플리케이션을 네트워크 서비스로 노출한다.<br>• 인그레스: 서비스를 http 또는 https 경로로 노출한다.<br>• 네트워크폴리시: OSI 3 또는 4 계층에서 트래픽 흐름을 제어한다. |
| 스토리지 | 스토리지 목록을 확인할 수 있으며 YAML/생성/수정/삭제할 수 있다.<br>• 퍼시스턴트볼륨: NFS 와 같이 스토리지 구현에 필요한 정보를 관리한다.<br>• 퍼시스턴트볼륨클레임: 스토리지에 대해 요청을 하는 리소스 정보를 관리한다.<br>• 스토리지클래스: 퍼시스턴트볼륨을 동적으로 배포한다. |
| 커스텀 리소스 | 쿠버네티스 커스텀 리소스를 관리한다. |
| 접근제어 | 쿠버네티스의 역할 기반 접근 제어를 관리하고 구성한다.<br>• 서비스어카운트: 주로 파드에 인증 정보를 제공한다. (사용자 역할)<br>• 롤/클러스터롤: API나 리소스에 대한 권한을 설정한다.<br>• 롤바인딩/클러스터롤바인딩: 롤/클러스터롤을 사용자 또는 서비스어카운트 등에 연결한다. |

| | |
|---|---|
| 모니터링 | 시스템 상태 및 각종 로그를 조회한다.<br>• 시스템: 클러스터 내 CPU, 메모리와 같은 리소스 사용량 정보를 제공한다.<br>• 이벤트 로그: 쿠버네티스 이벤트 로그를 조회한다.<br>• 컨테이너 로그: 쿠버네티스에 배포되었던 컨테이너에서 발생한 로그를 조회한다.<br>• 감사 로그: 클러스터 내 감사 로그를 조회한다.<br>• 알림 로그: 클러스터 내 발생한 알림 정보를 조회한다.<br>• 서비스메시: 클러스터 내 서비스 간의 통신 정보를 표시한다. |
| 설정 | 클러스터를 사용하는 멤버를 구성하고 알림 정책을 세운다. 레지스트리 정보 및 알림을 위한 클러스터 수신자 정보를 설정한다.<br>• 클러스터 멤버: 클러스터 권한과 사용자/그룹을 연결한다.<br>• 클러스터 수신자: 알림 발송을 받을 클러스터 수신자 정보를 관리한다.<br>• 알림 정책: 클러스터 알림 발송 규칙 및 발송 주기/수신자 등을 설정힌다.<br>• 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 | 헬름과 카탈로그를 이용해 애플리케이션을 클러스터에 배포한다.<br>• 헬름: 헬름 차트로 애플리케이션을 배포한다. 주로 빌드를 수행하지 않는 서비스형 애플리케이션을 배포한다.<br>• 카탈로그: 카탈로그로 애플리케이션을 배포한다. 빌드 파이프라인을 지원한다. |
| 빌드 | 애플리케이션 빌드에 필요한 정보를 제어하고 관리한다. 컨테이너 이미지를 빌드할 수 있고 빌드에 필요한 파이프라인 또는 승인에 대해 설정한다.<br>• 파이프라인: 태스크를 조합하여 빌드를 수행할 파이프라인을 작성한다.<br>• 승인: 파이프라인에서 발생한 승인 요청을 처리한다.<br>• 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 | 쿠버네티스의 역할 기반 접근 제어를 관리하고 구성한다.<br>• 서비스어카운트: 주로 파드에 인증 정보를 제공한다. (사용자 역할)<br>• 롤: API나 리소스에 대한 권한을 설정한다.<br>• 롤바인딩: 롤을 사용자 또는 서비스어카운트 등에 연결한다. |
| 모니터링 | 시스템 및 애플리케이션 상태 및 각종 로그를 조회한다.<br>• 시스템: 네임스페이스 내 CPU, 메모리와 같은 리소스 사용량 정보를 제공한다.<br>• 애플리케이션: 스카우터로 수집한 애플리케이션에 대한 메트릭 정보를 제공한다.<br>• 이벤트 로그: 네임스페이스에서 발생한 쿠버네티스 이벤트 로그를 조회한다.<br>• 컨테이너 로그: 네임스페이스에 배포되었던 컨테이너에서 발생한 로그를 조회한다.<br>• 감사 로그: 네임스페이스 내 감사 로그를 조회한다.<br>• 알림 로그: 네임스페이스 내 발생한 알림 정보를 조회한다. |

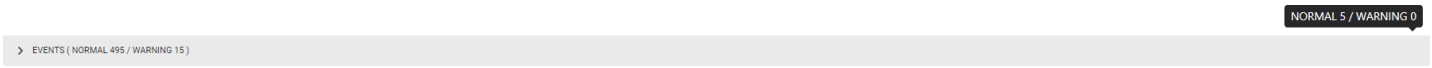
| | |
|---|---|
| 설정 | 네임스페이스를 사용하는 멤버를 구성한다. 그리고 알림 정책을 세우고 알림을 위한 네임스페이스 수신자 정보를 설정한다.<br>• 네임스페이스 멤버: 네임스페이스 권한과 사용자/그룹을 연결한다.<br>• 네임스페이스 수신자: 알림 발송을 받을 네임스페이스 수신자 정보를 관리한다.<br>• 알림 정책: 네임스페이스 알림 발송 규칙 및 발송 주기/수신자 등을 설정한다. |

### 3.1.3. 이벤트 영역

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

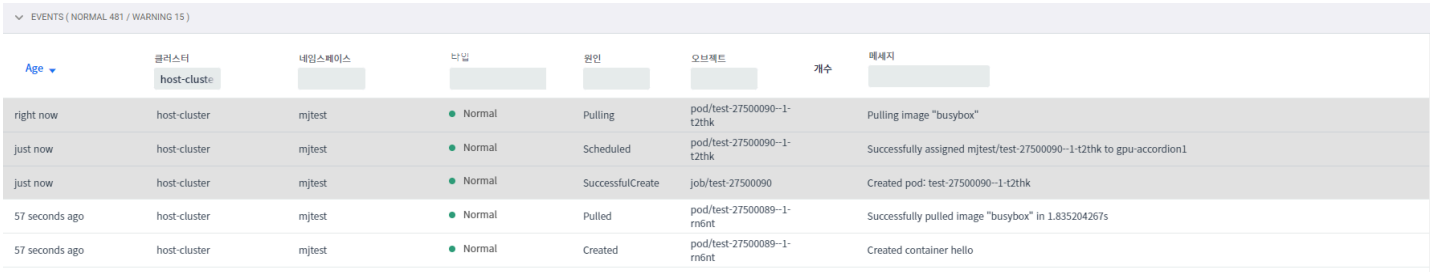
#### 3.1.3.1. 이벤트 영역의 최소화

이벤트 영역에 우측 툴팁으로 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.




| 항목 | 설명 |
| --- | --- |
| Age | 이벤트가 발생한 시간 |
| 클러스터 | 클러스터 명 |
| 네임스페이스 | 네임스페이스 명 |
| 타입 | 이벤트 타입<br>• Normal: 일반적인 작업으로 발생된 이벤트<br>• Warning: 오류에 의해 발생된 이벤트 |
| 원인 | 이벤트 발생 이유 |
| 오브젝트 | 오브젝트 정보 |
| 메세지 | 이벤트 상세 메시지 |

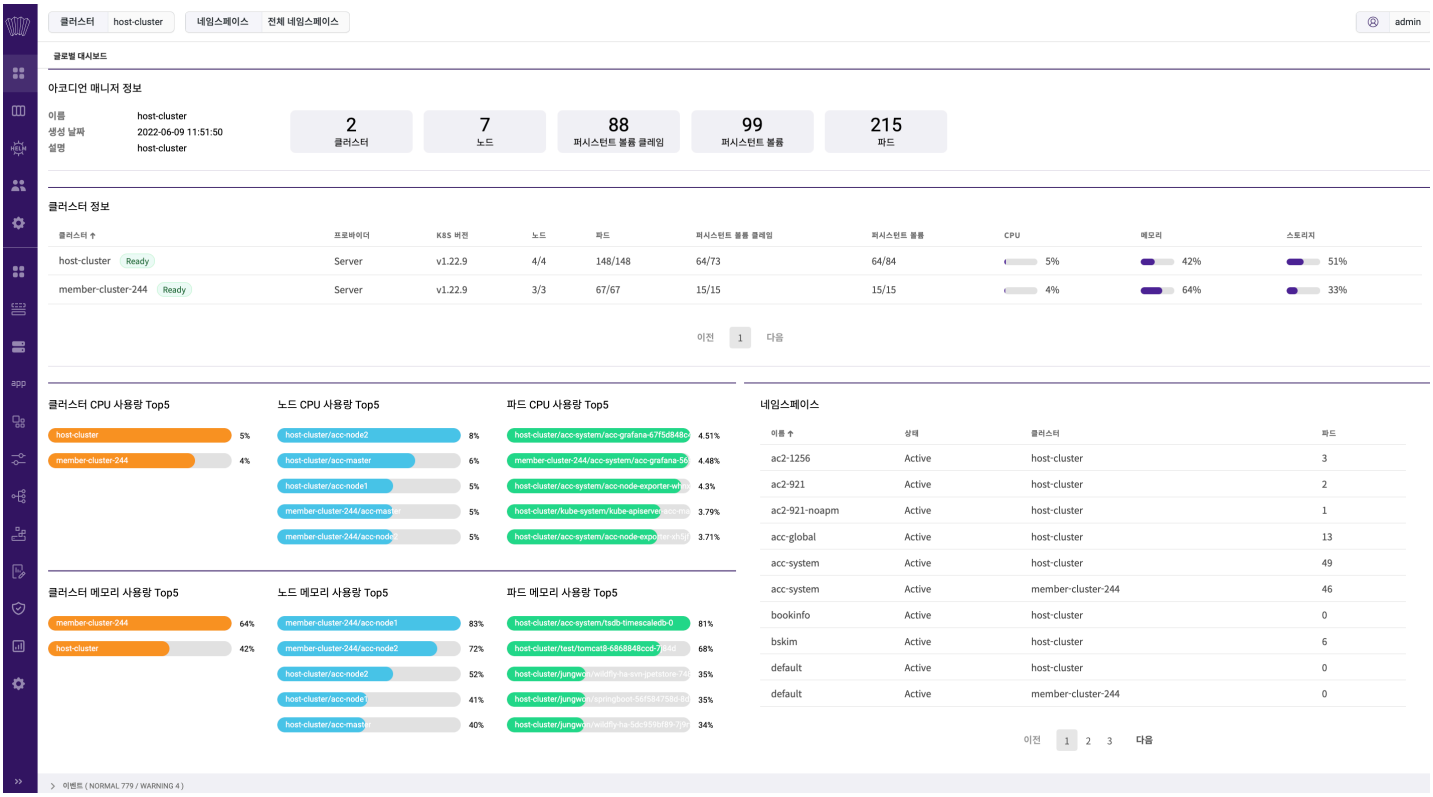
# 4. 메뉴

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

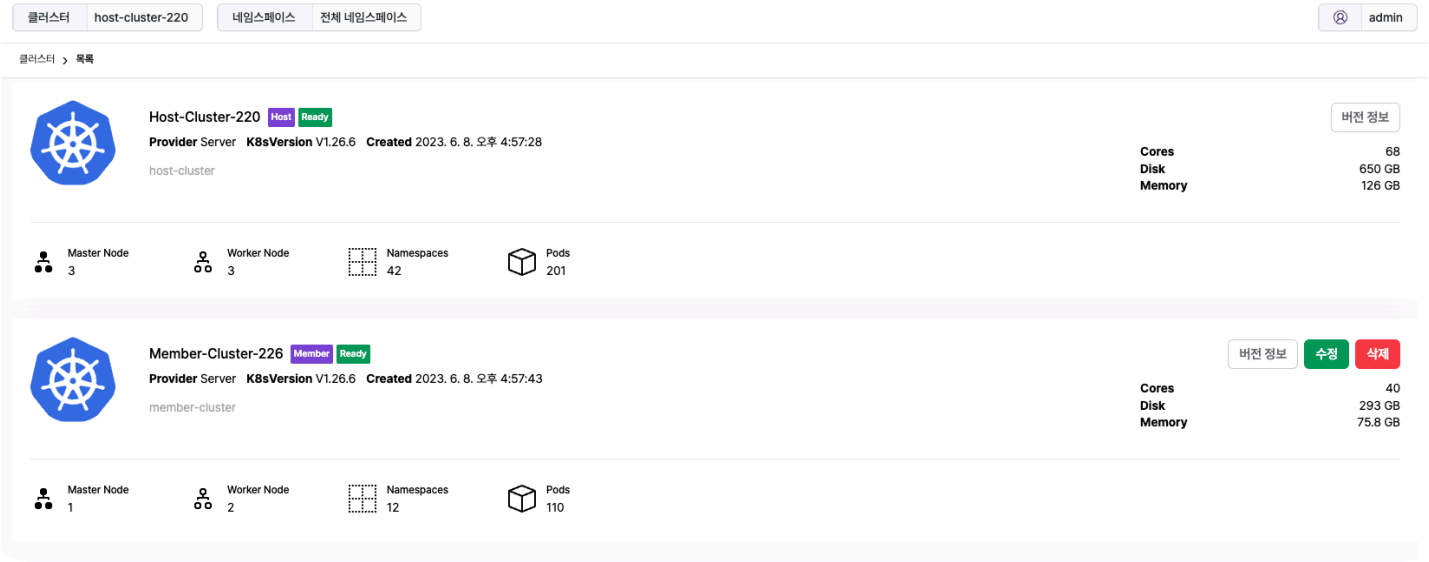
클러스터는 컨테이너화된 애플리케이션을 실행하기 위한 노드의 집합을 의미한다. 컨테이너화된 애플리케이션은 개별 노드에 얽매이지 않고 클러스터에 배포된다. 클러스터는 클러스터를 관리하는 호스트 클러스터와 사용자의 애플리케이션이 배포되는 멤버 클러스터로 구분되며 호스트 클러스터는 멤버 클러스터의 역할도 수행한다. 아코디언은 하나의 호스트 클러스터와 하나 이상의 멤버 클러스터로 구성된다.

**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

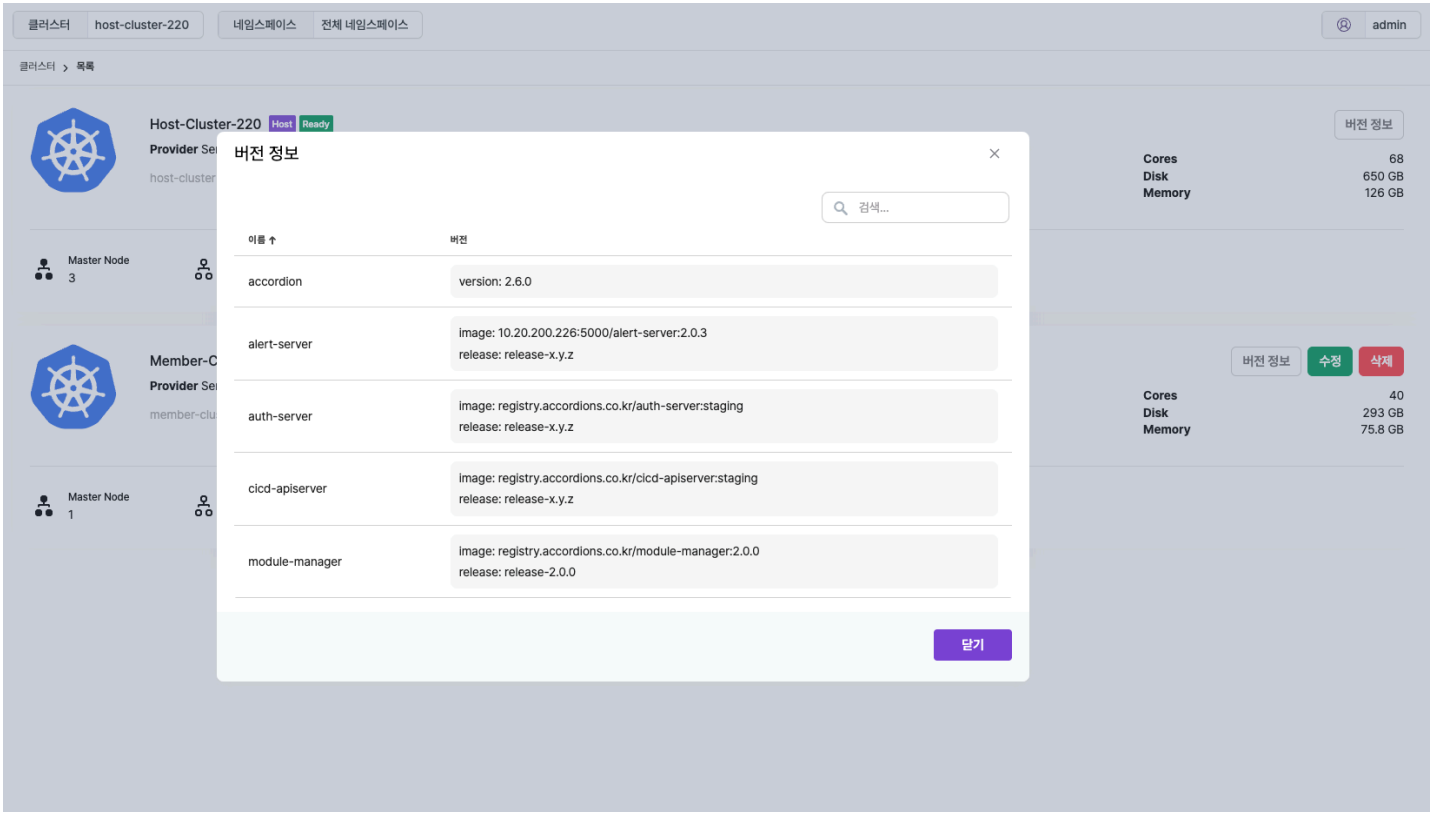



클러스터 메뉴에서는 호스트 클러스터의 정보를 조회하고 멤버 클러스터를 관리할 수 있다.

| 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 |

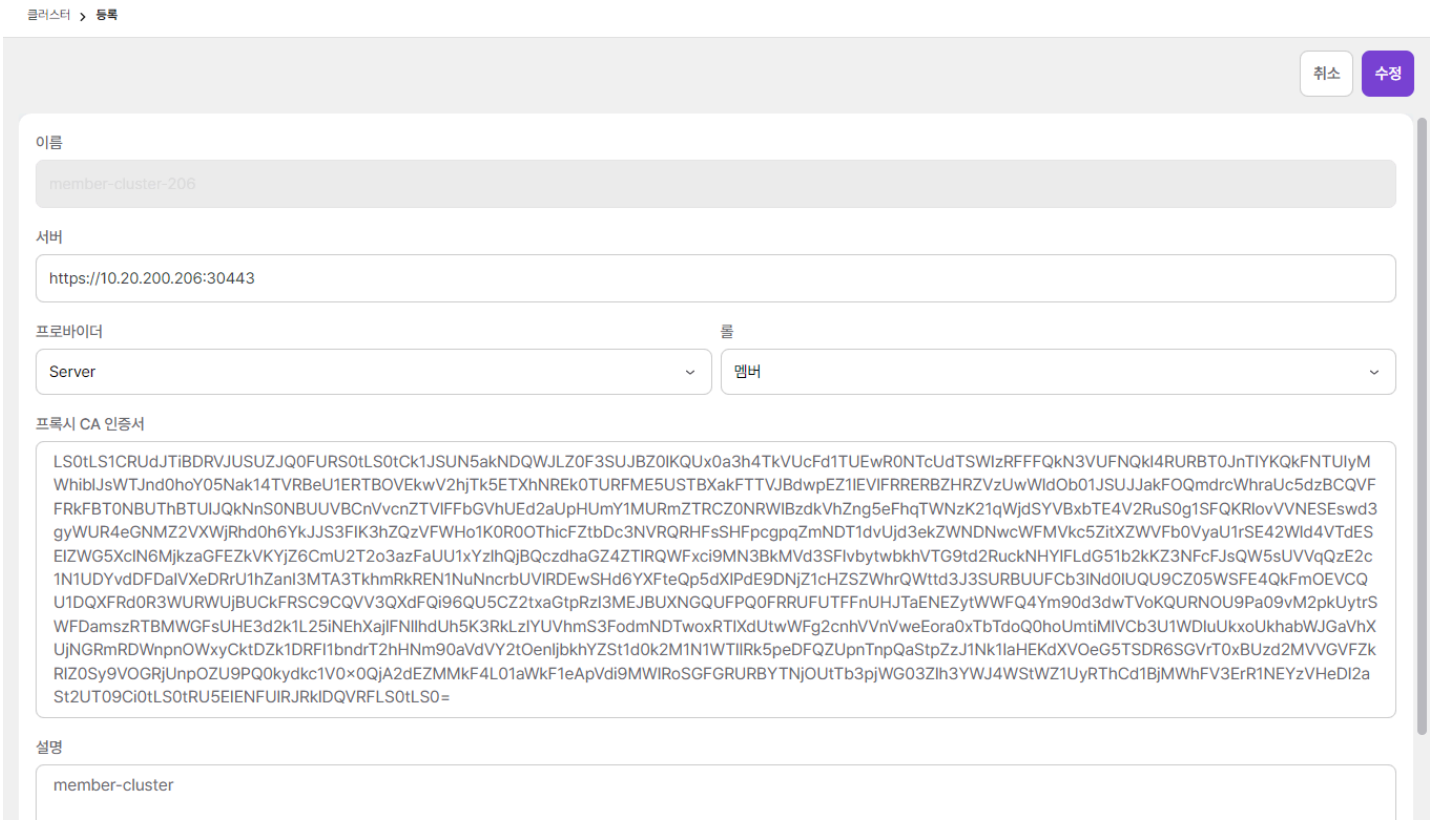
## 버전 정보

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.

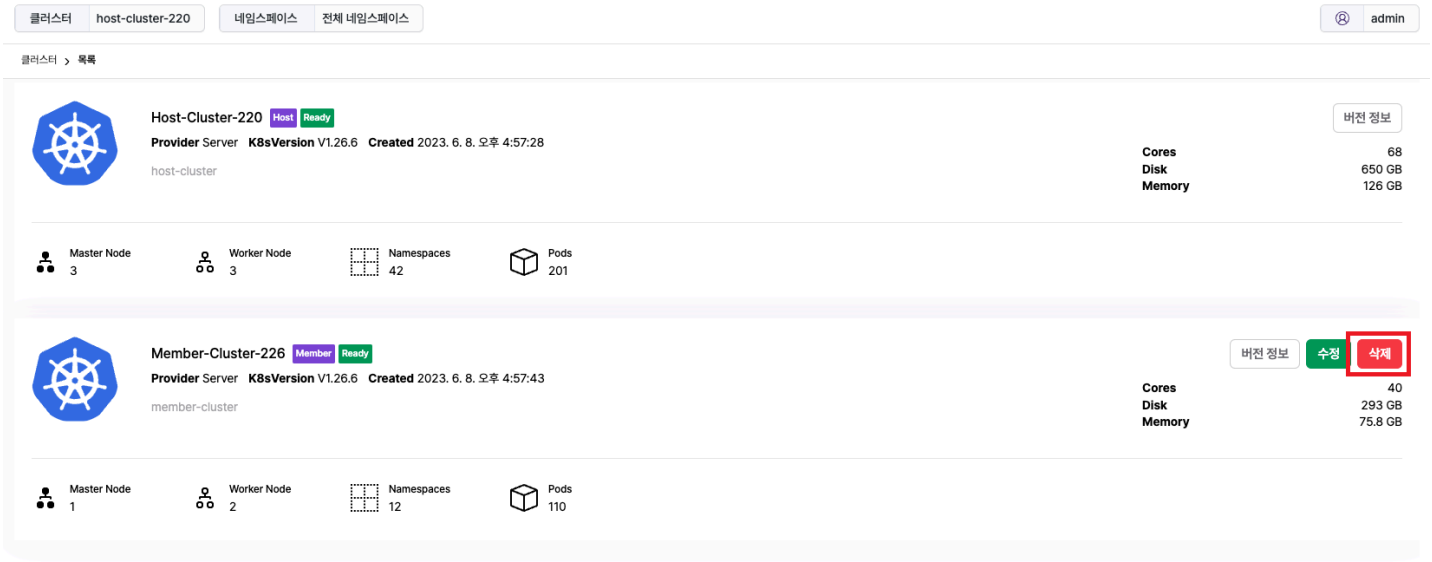


제공하는 정보는 다음과 같다.

| 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 것이지 실제 클러스터를 삭제하지는 않는다.



### 4.1.3. 헬름

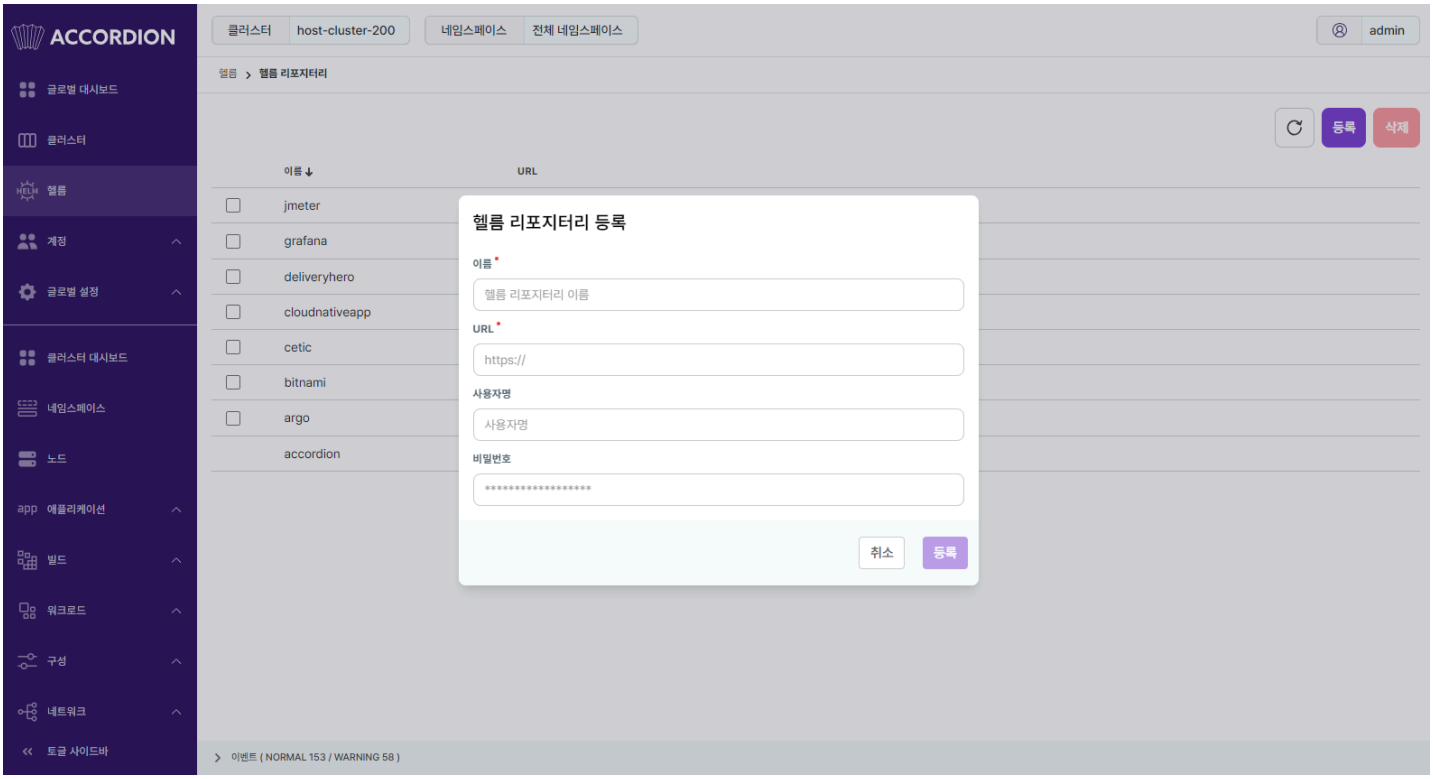
헬름은 쿠버네티스 애플리케이션 배포를 지원하는 도구로 아코디언에서는 헬름 리포지터리 및 차트를 관리하고 이를 이용해 클러스터에 애플리케이션을 배포할 수 있게 서비스를 제공한다.

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

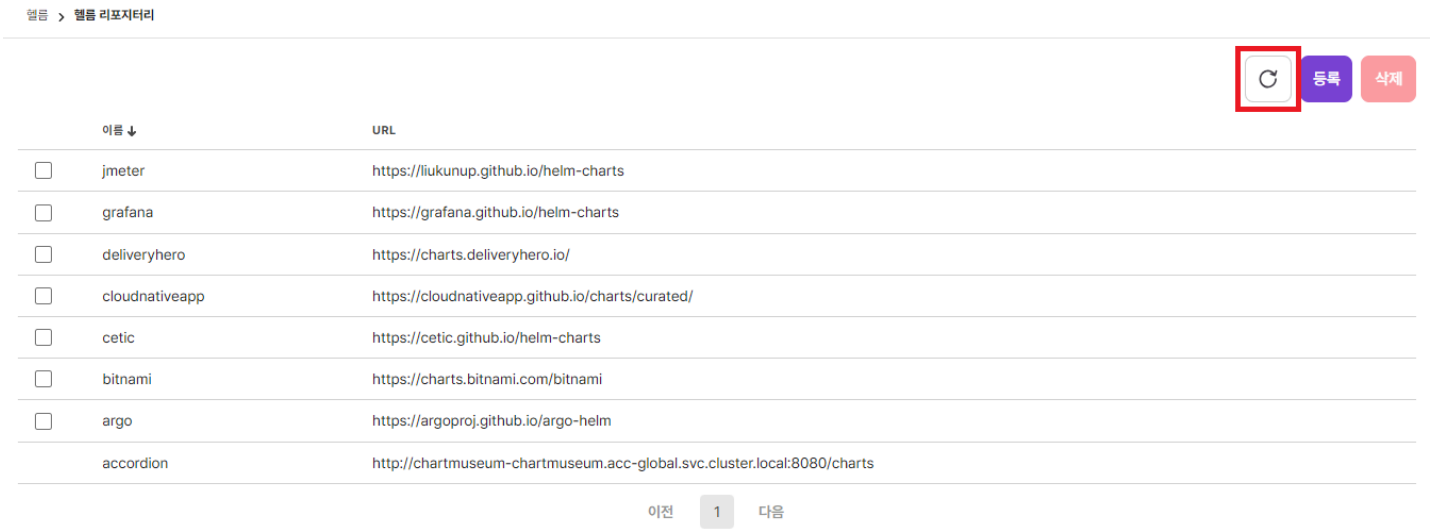
헬름 메뉴에 접근하여 상단의 `등록` 버튼을 선택하면 헬름 리포지터리 정보를 입력하는 모달 화면이 표시된다.




| 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

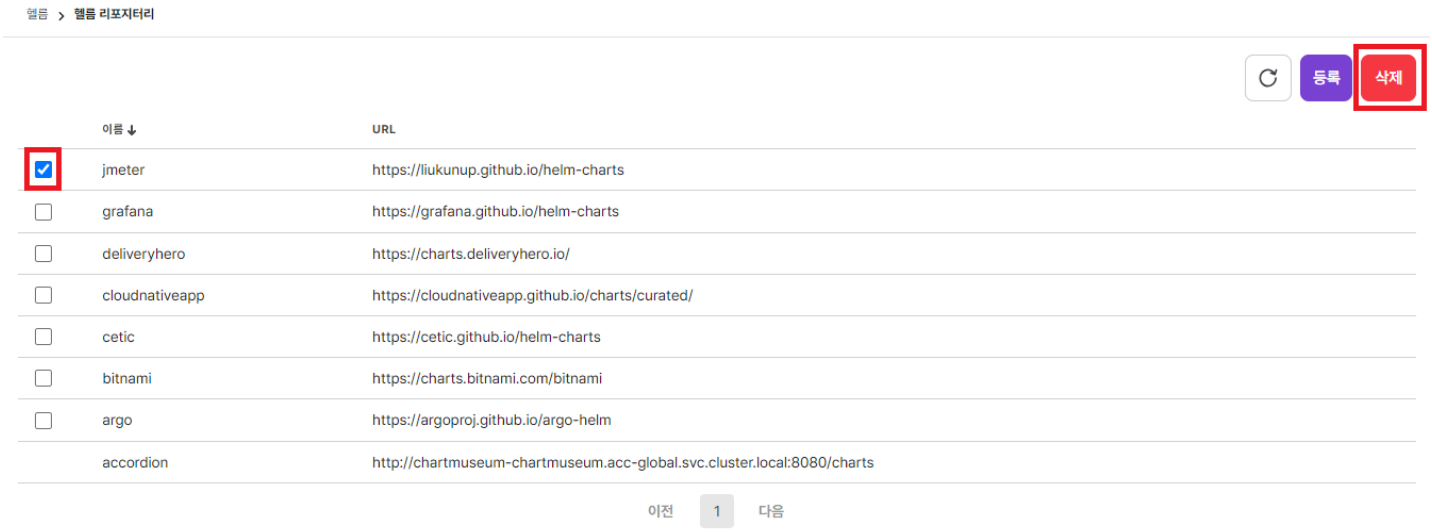
우측 상단의 업데이트 버튼을 선택하면 헬름 리포지터리의 차트 정보를 최신으로 업데이트할 수 있다.




## 헬름 리포지터리 삭제

삭제하려는 리포지터리를 선택하고 우측 상단의 `삭제` 버튼을 선택한다.

**NOTE** 기본 제공하는 `accordion` 리포지터리는 삭제할 수 없다.

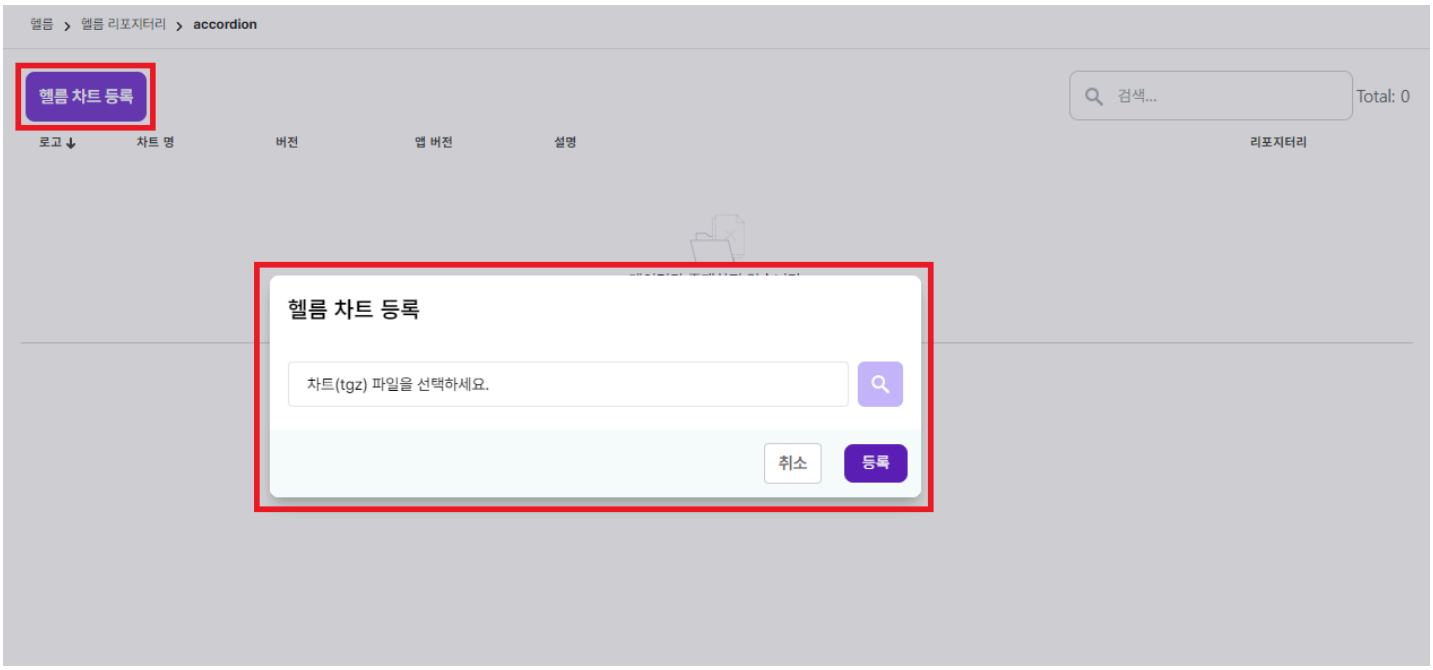


## 헬름 차트

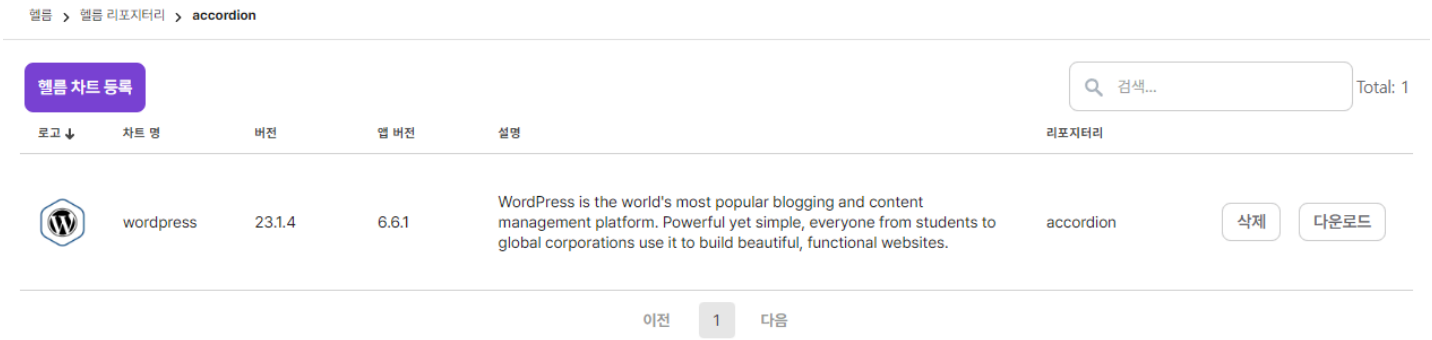
### 헬름 차트 등록

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.

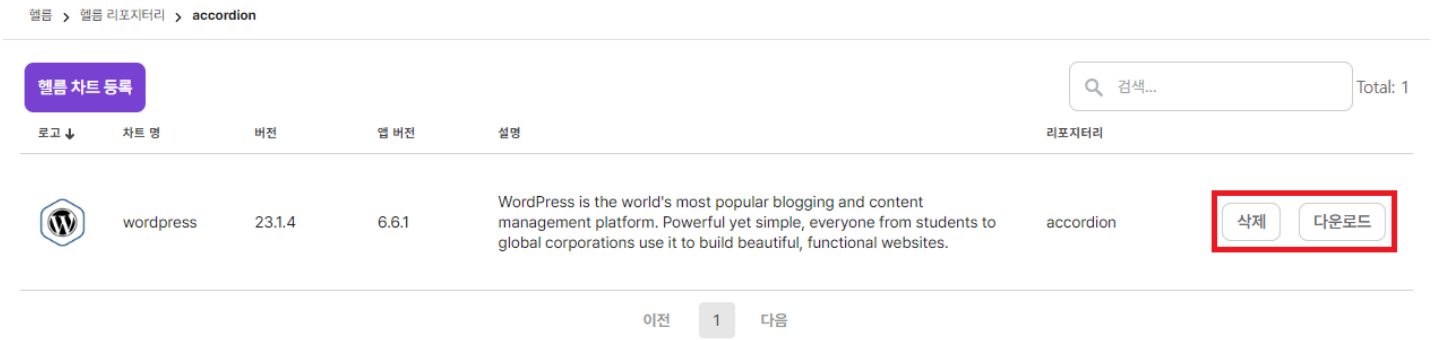



헬름 차트가 정상적으로 등록되면 아래와 같이 조회할 수 있다.




### 헬름 차트 삭제와 다운로드

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

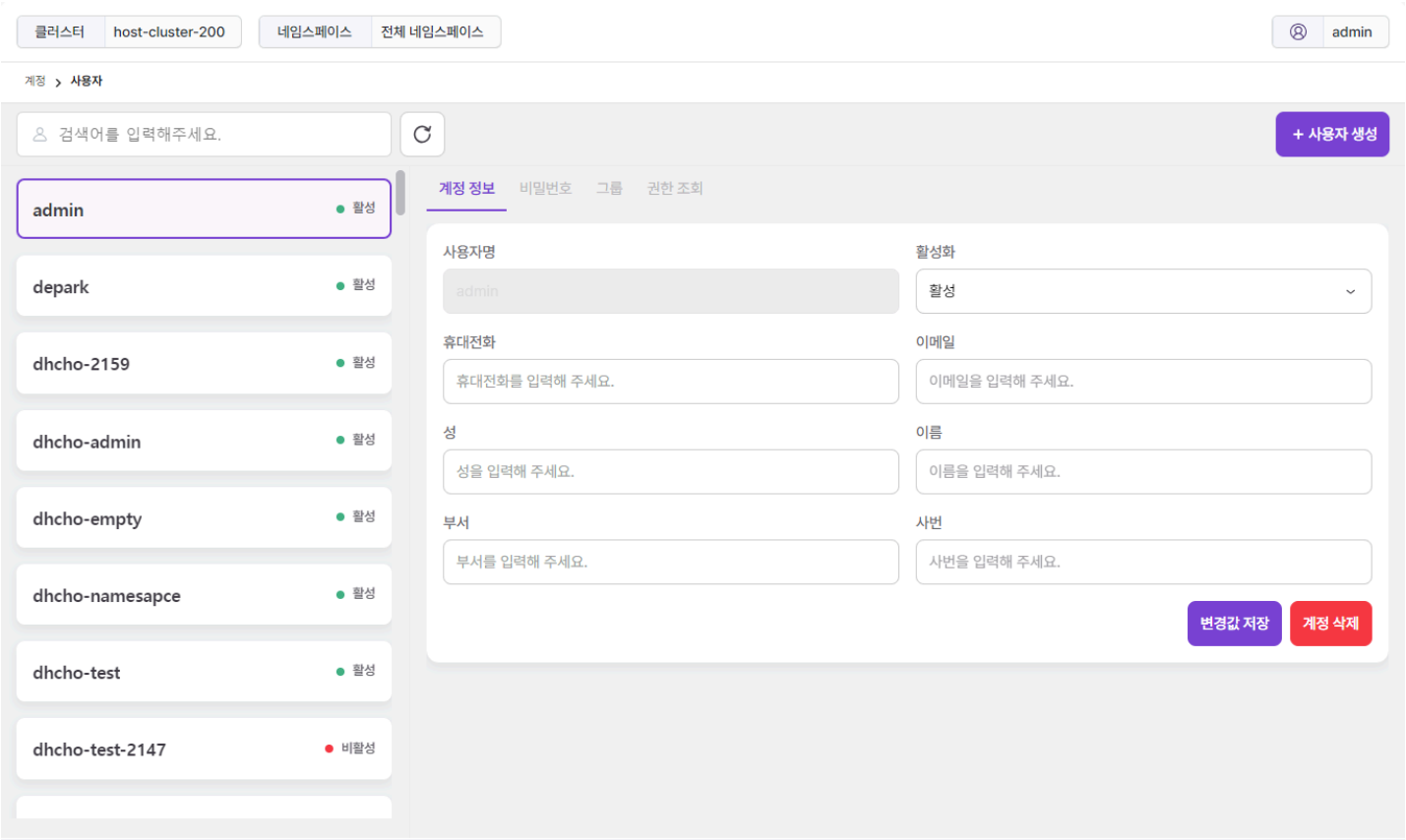


### 4.1.4. 계정

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) |

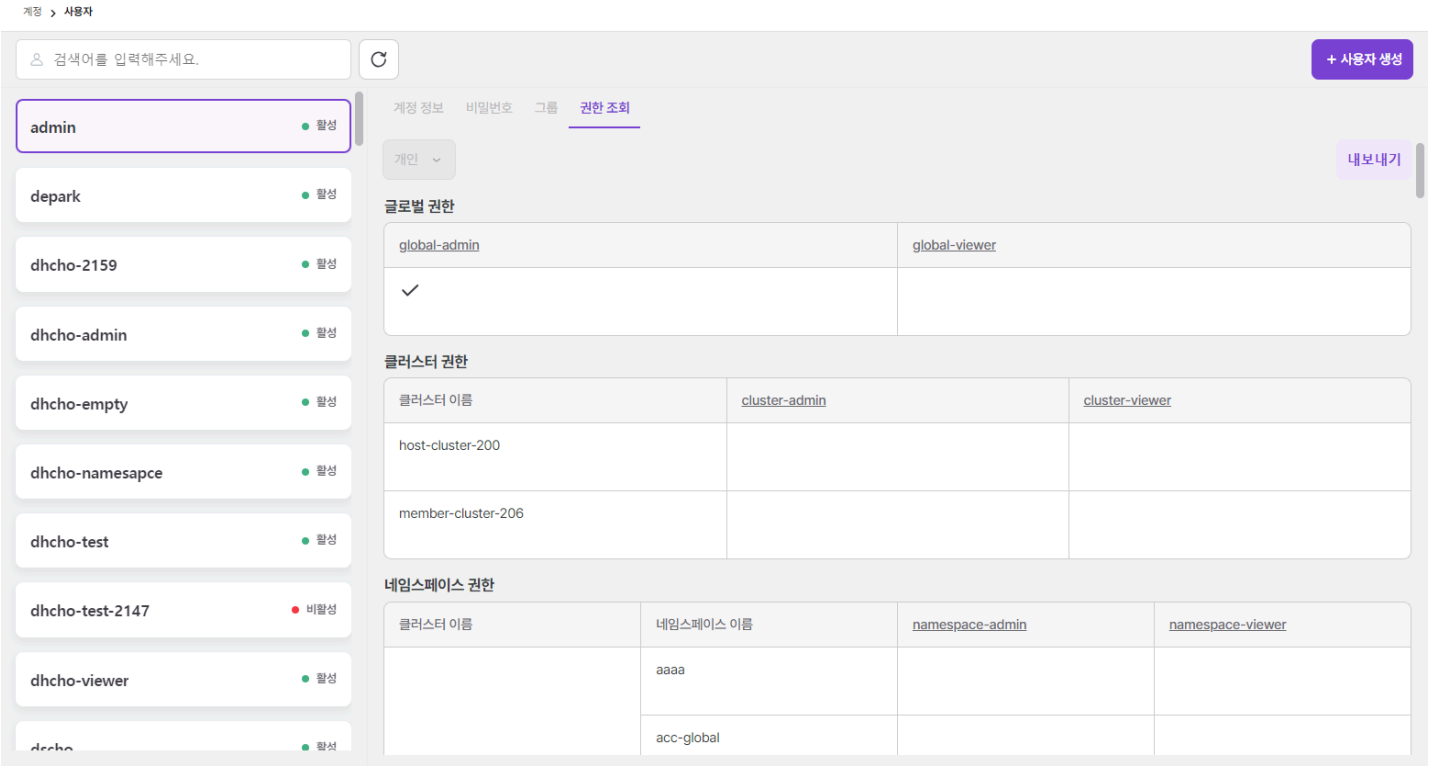
## 비밀번호

| 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

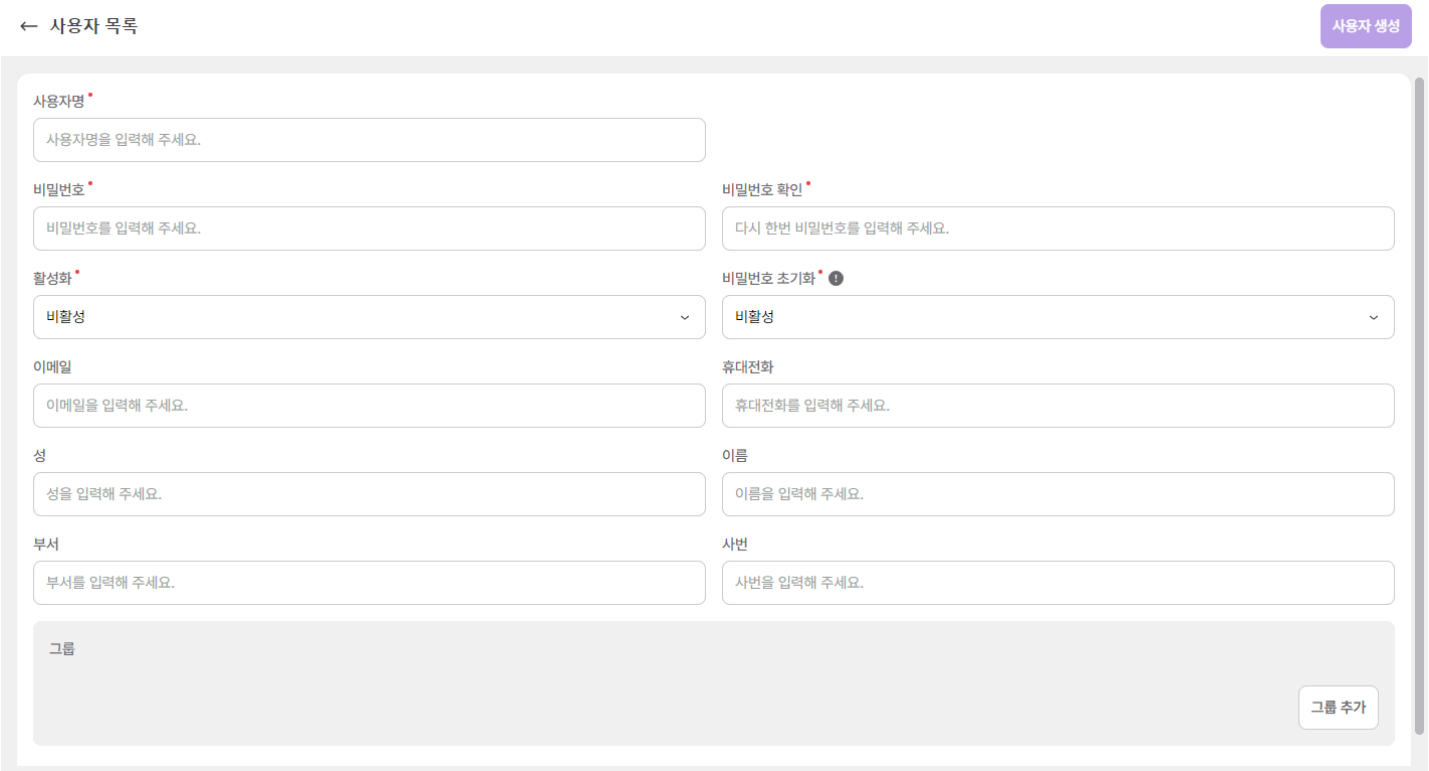



| 항목 | 설명 |
| --- | --- |
| 권한 조회 유형 | 전체 (개인+그룹), 개인 , 그룹 으로 구성되었으며, 특정 유형을 선택하여 조회할 수 있다. |
| 내보내기 | 권한을 csv 파일로 내보낸다. |
| 글로벌 권한<br>클러스터 권한<br>네임스페이스 권한 | 사용자가 해당 권한을 가지고 있다면 v 로 체크된다.<br>상단의 권한 이름을 선택할 경우 해당 권한 페이지로 이동된다. |

## 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 수정

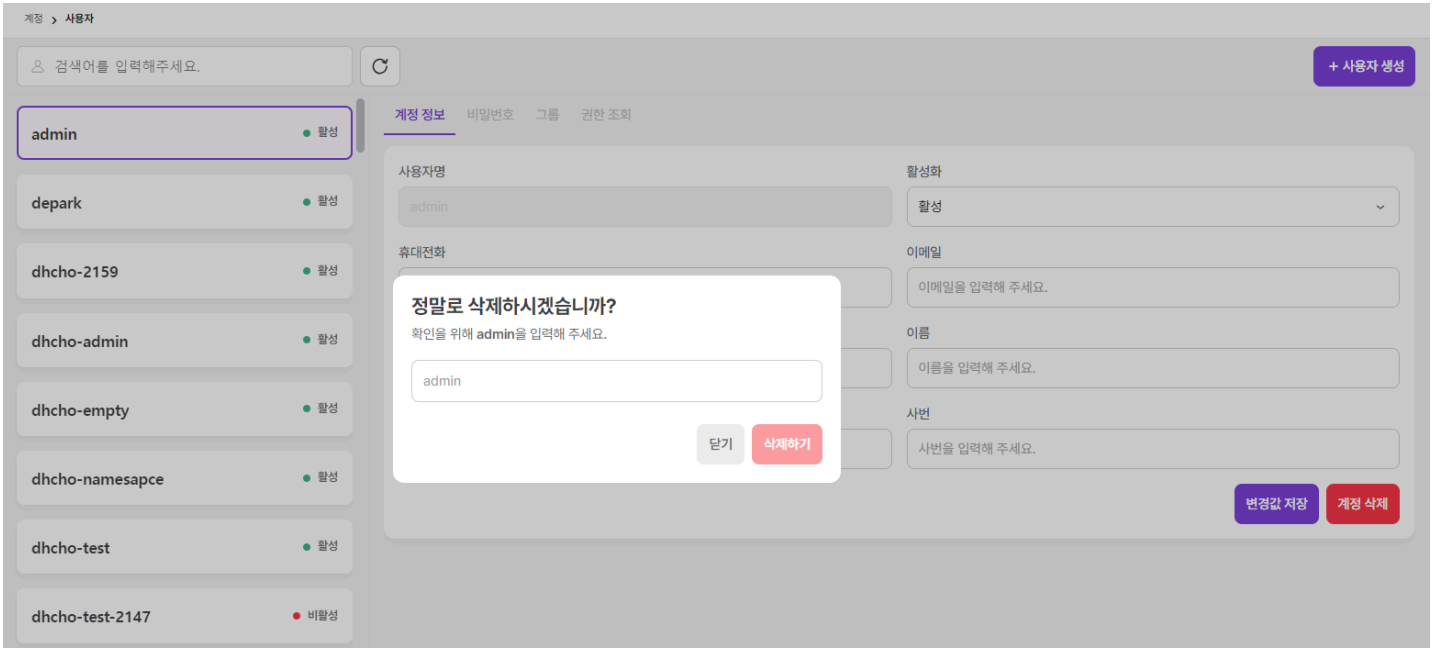
사용자를 수정하기 위해서는 사용자 목록에서 사용자를 선택하고, 이후 수정할 정보를 입력한 후 `변경값 저장` 버튼을 선택한다.

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.

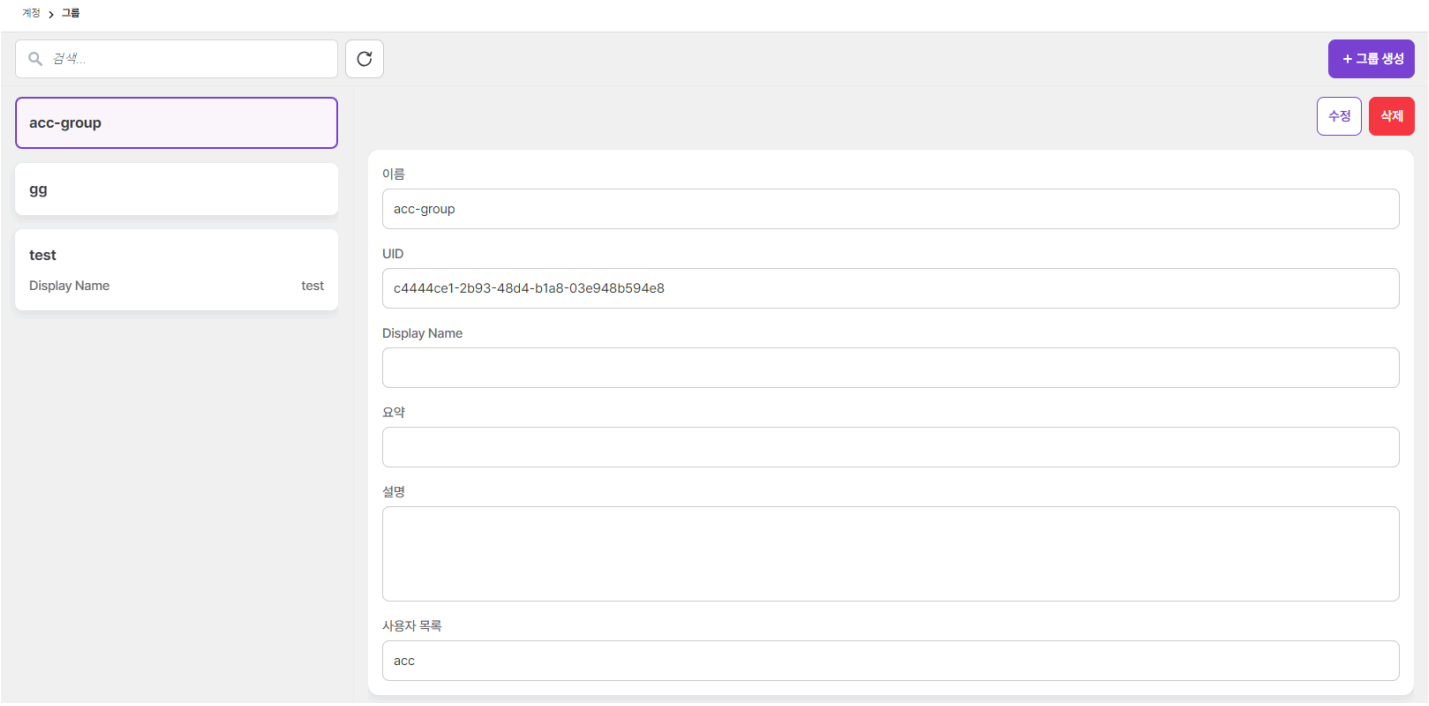
이후 나타나는 모달에서 사용자명을 입력한 후 `삭제하기` 버튼을 선택하면 완료된다.



### 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

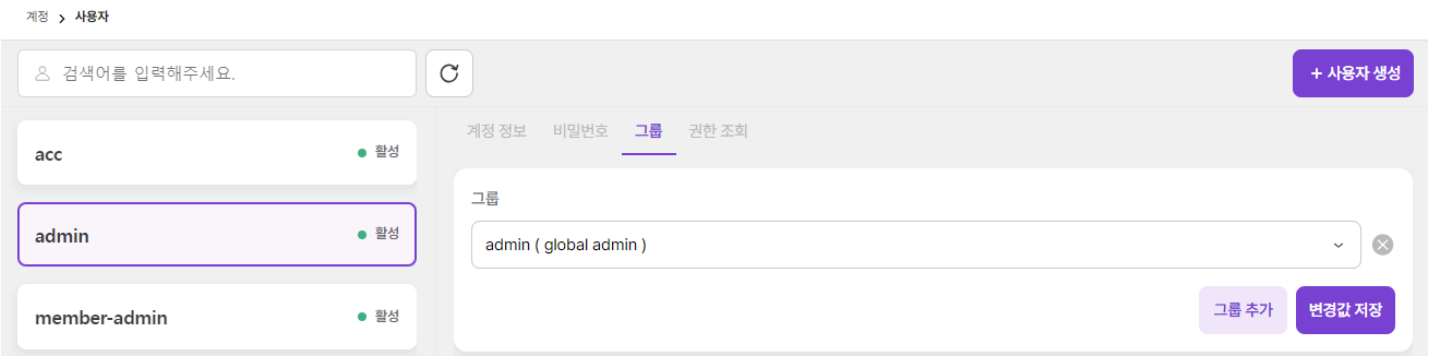
그룹을 등록하기 위해서는 우측 상단의 + 그룹 생성 버튼을 선택한 후 필수 항목인 이름 및 기타 선택항목을 작성 후 그룹 생성 버튼을 선택한다.

← 그룹 목록
그룹 생성
이름*
Display Name
요약
설명

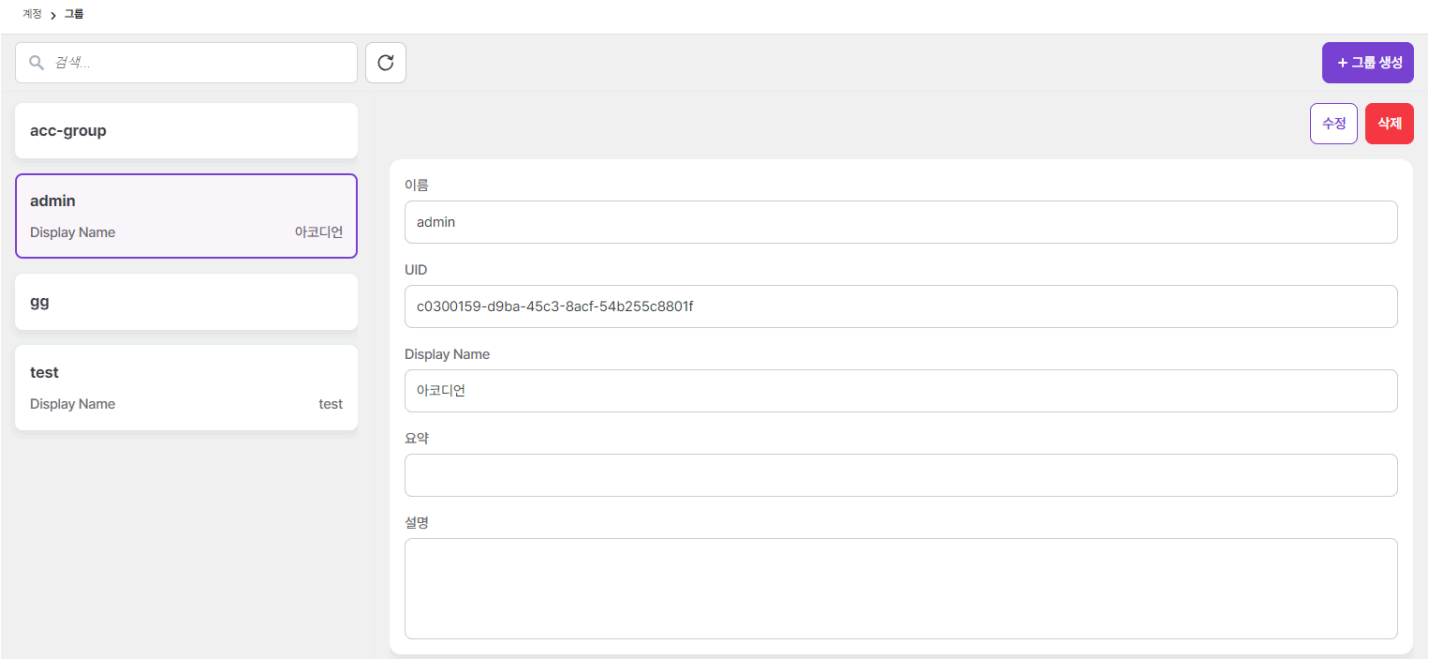
### Display Name

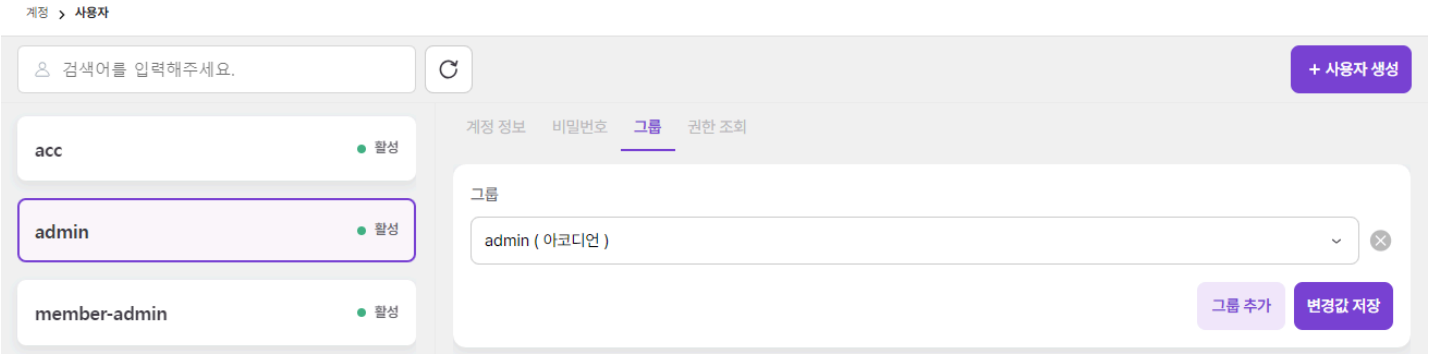
그룹을 등록하는 곳에서 실제로 표시될 이름을 정의한다.

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



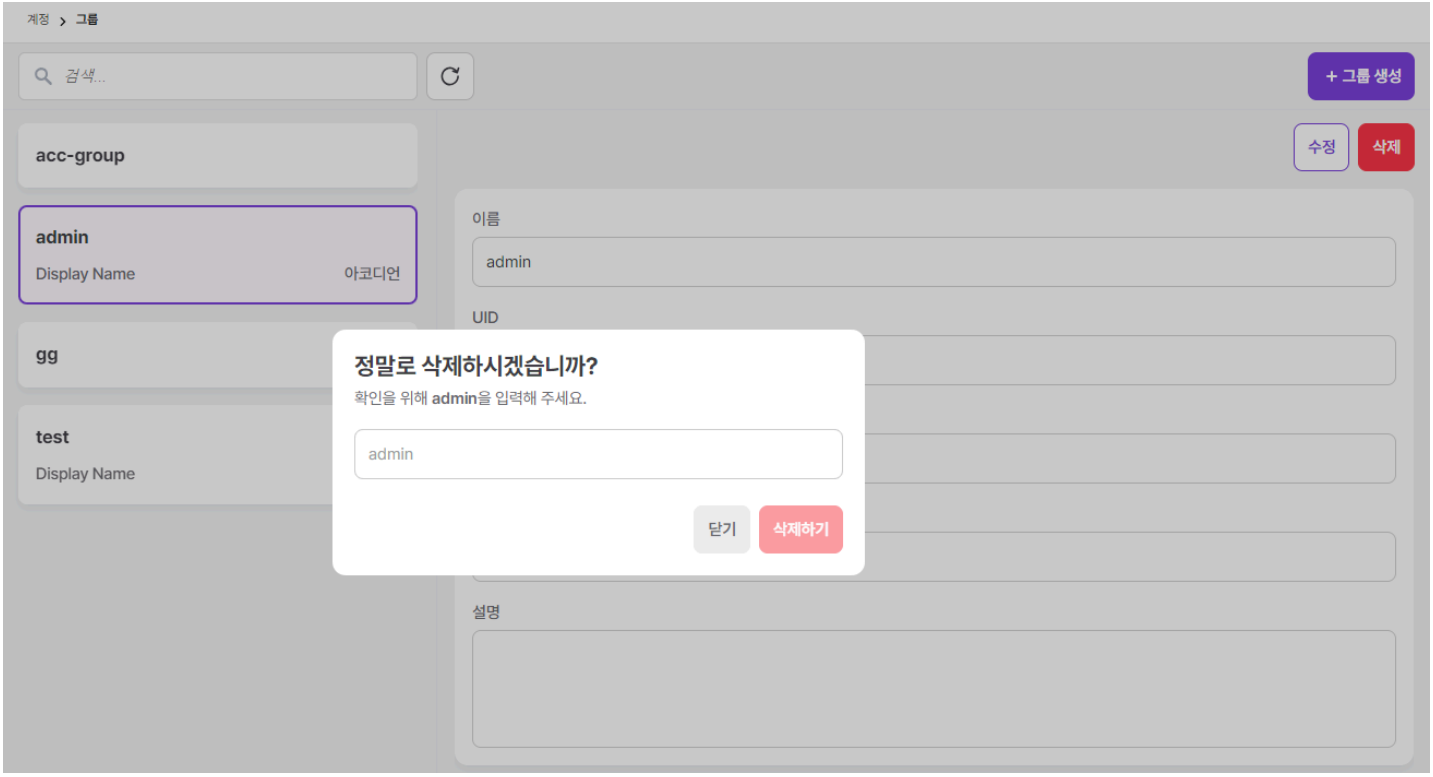

위의 이미지와 같이 그룹명 ( Display Name ) 으로 출력된다.
등록된 그룹의 정보를 수정하여 `Display Name` 에 정보를 수정하면 해당 내용으로 출력된다.



## 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.



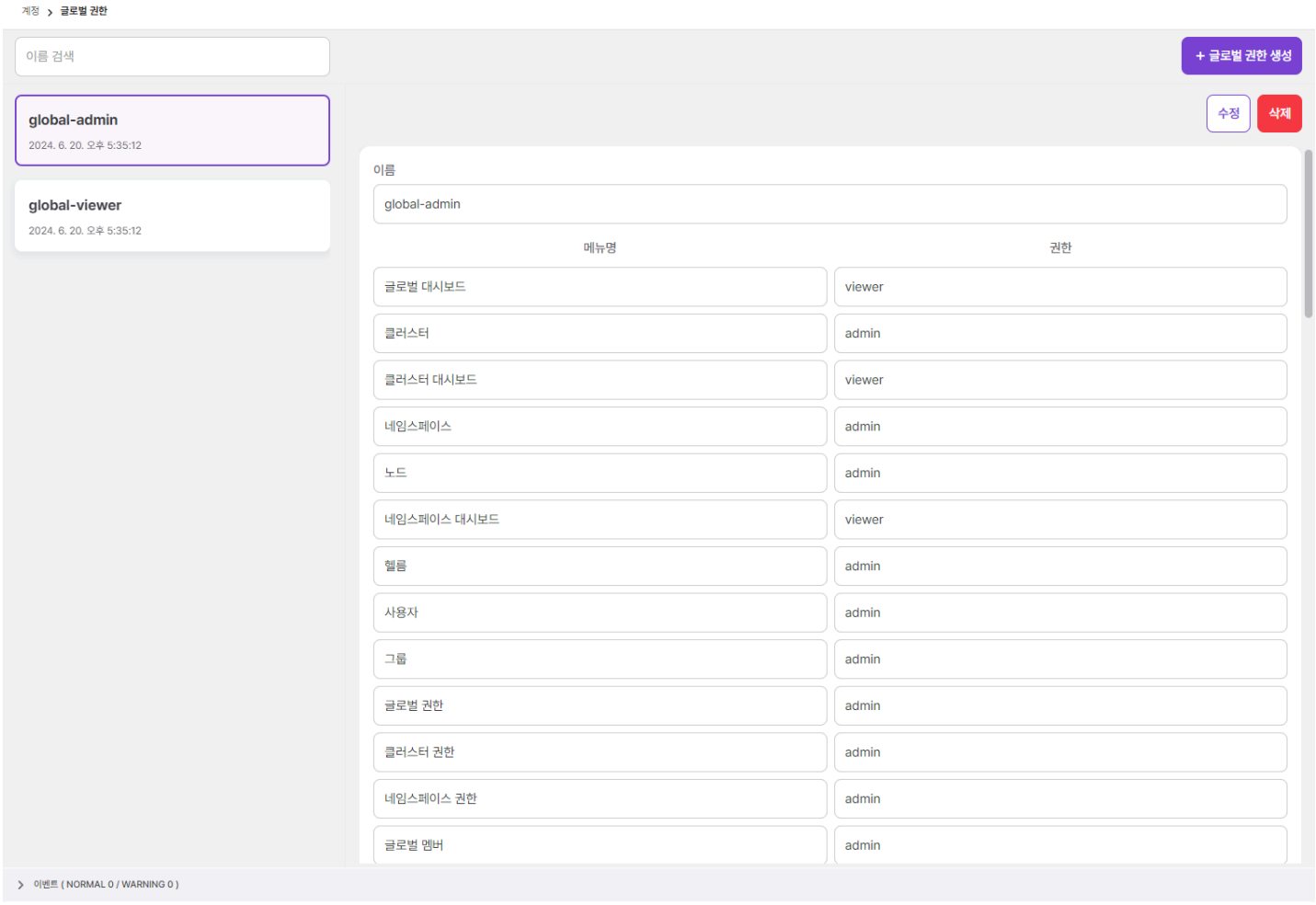
### 4.1.4.3. 권한

권한은 여러 메뉴와 롤의 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 롤 설정 가능 (글로벌 스코프)
  - 네임스페이스 메뉴에 대한 롤 설정 가능 (클러스터 스코프)
  - 카탈로그 메뉴에 대한 롤 설정 가능 (네임스페이스 스코프)
- 클러스터 권한
  - 네임스페이스 메뉴에 대한 롤 설정 가능 (클러스터 스코프)
  - 카탈로그 메뉴에 대한 롤 설정 가능 (네임스페이스 스코프)
- 네임스페이스 권한
  - 카탈로그 메뉴에 대한 롤 설정 가능 (네임스페이스 스코프)

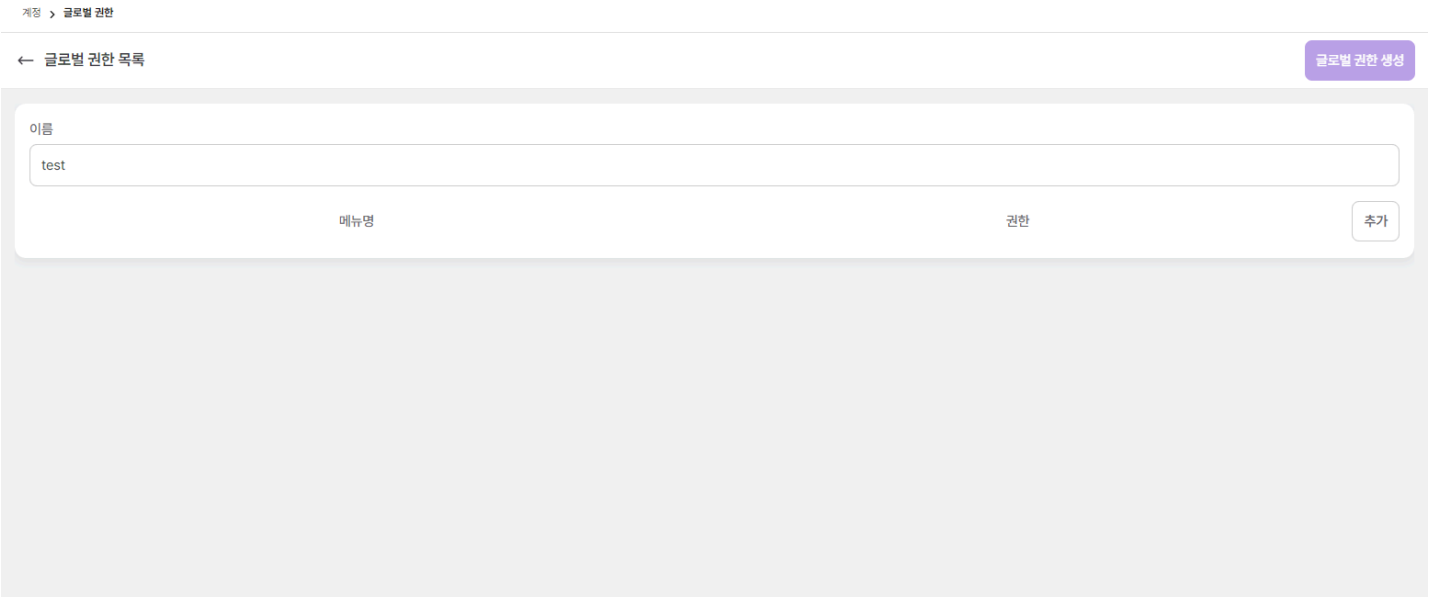
**TIP** 본문에서는 글로벌 권한을 예로 설명한다.

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



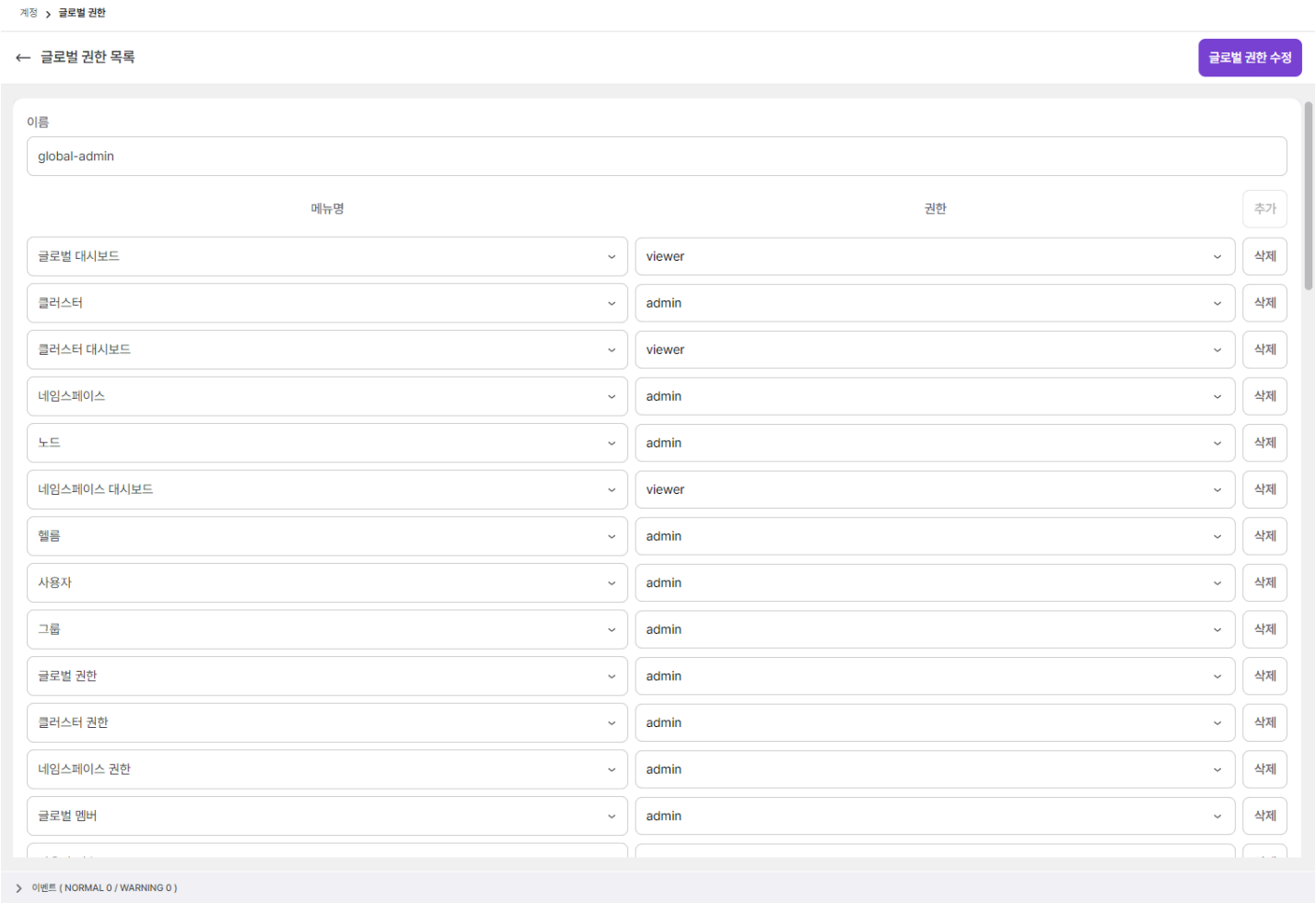
## 권한 등록

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



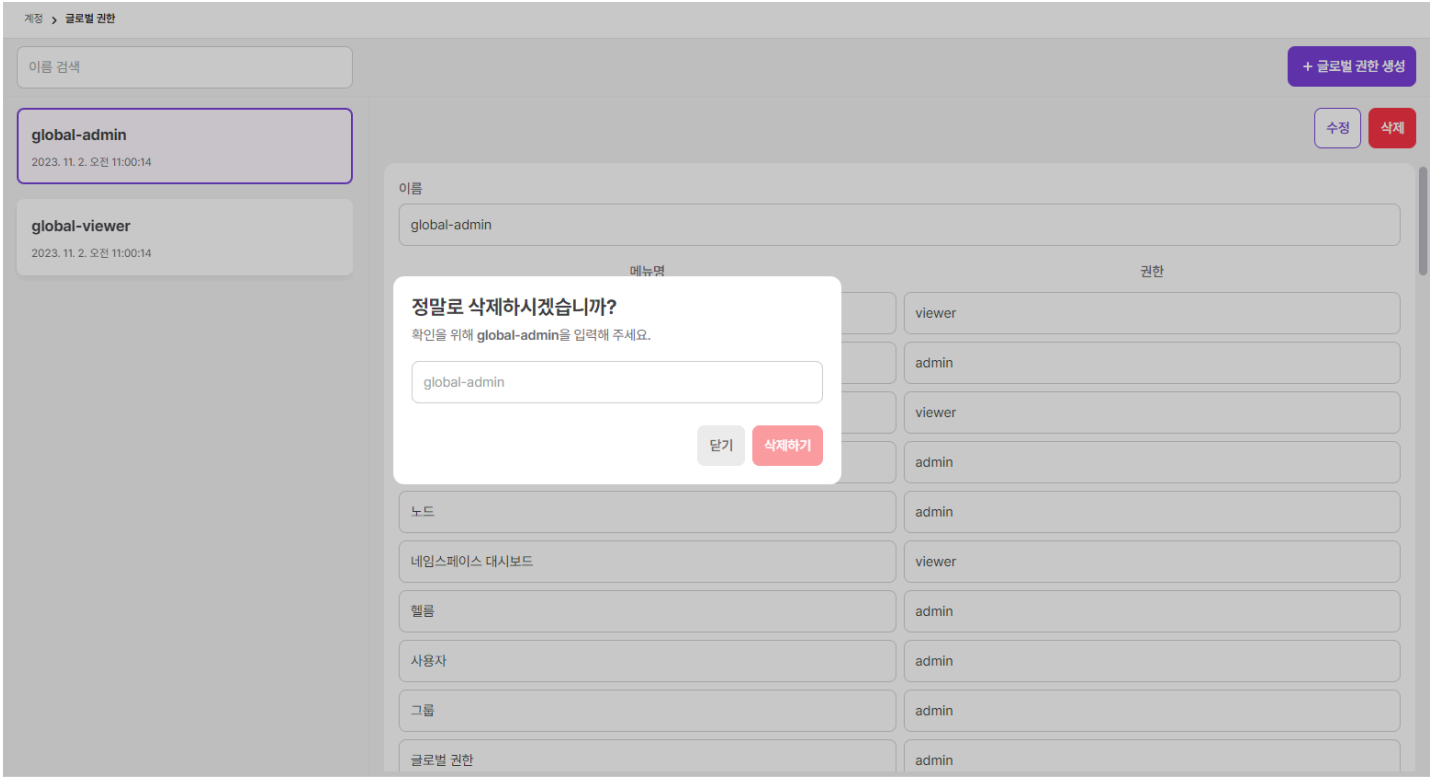

## 권한 수정

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



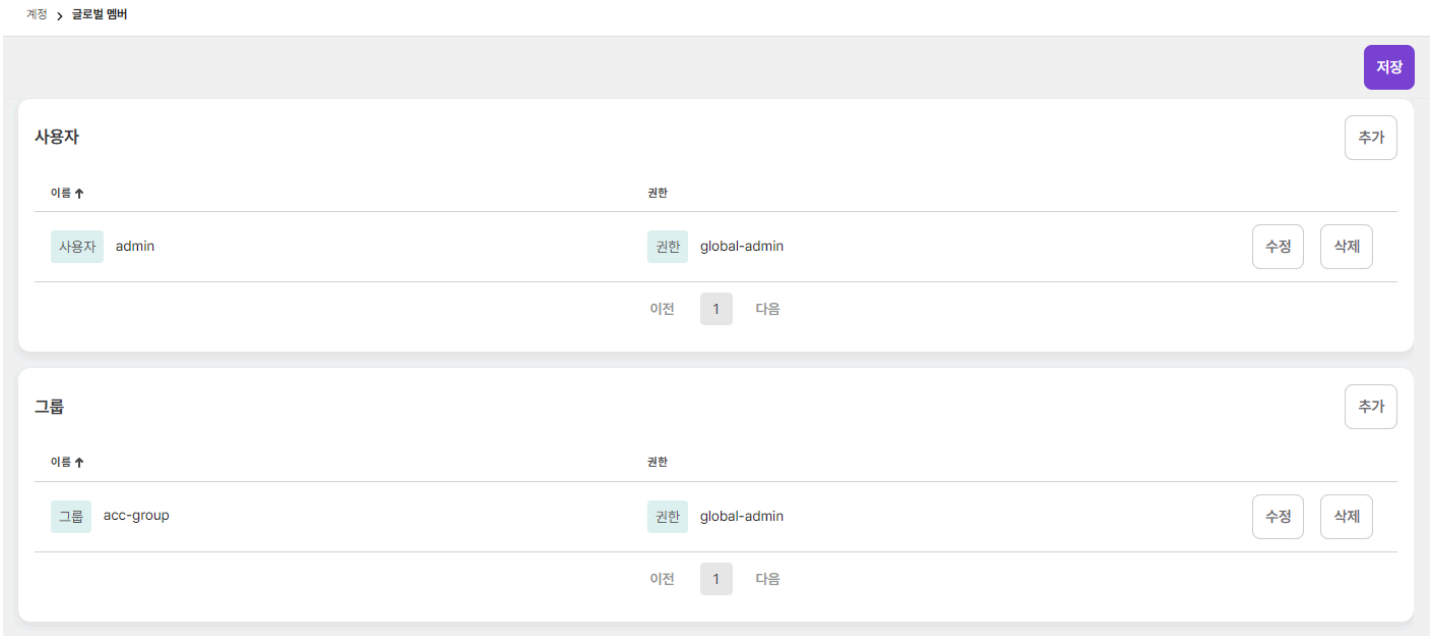
## 권한 삭제

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

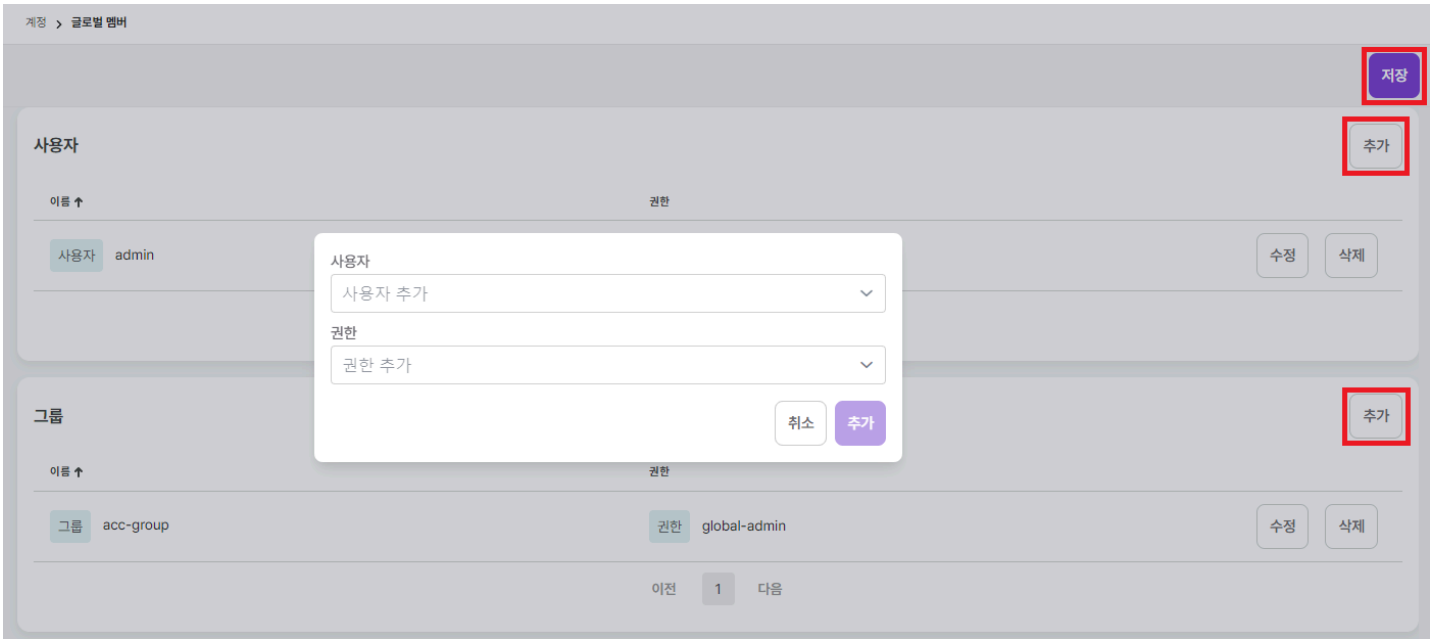
글로벌 멤버는 사용자와 그룹의 글로벌 권한을 관리한다.




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

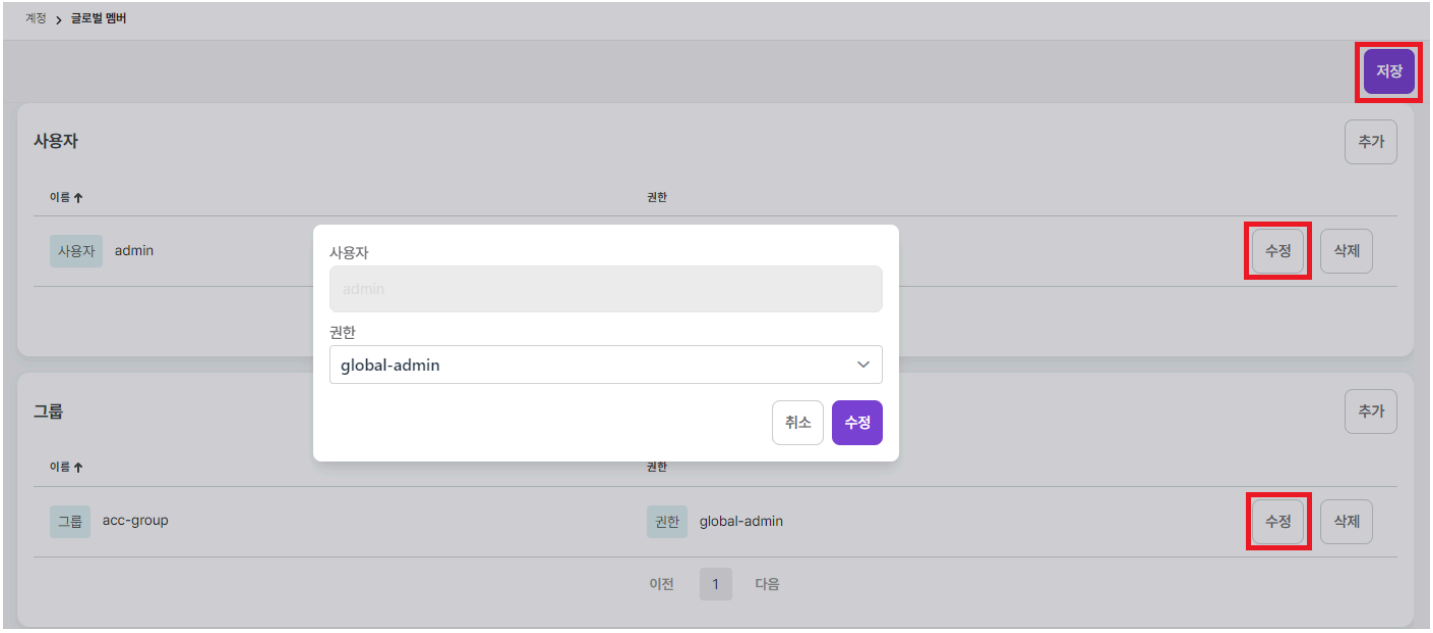
#### 멤버 추가

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.



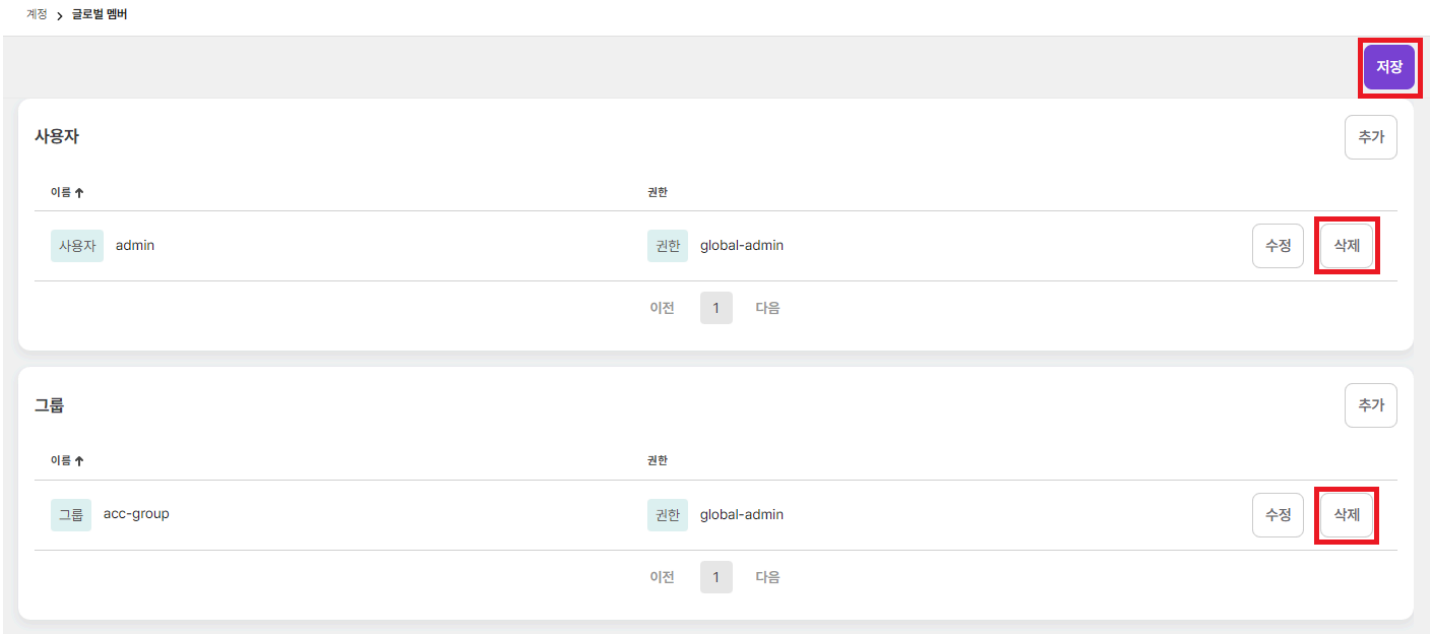
## 멤버 권한 수정

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



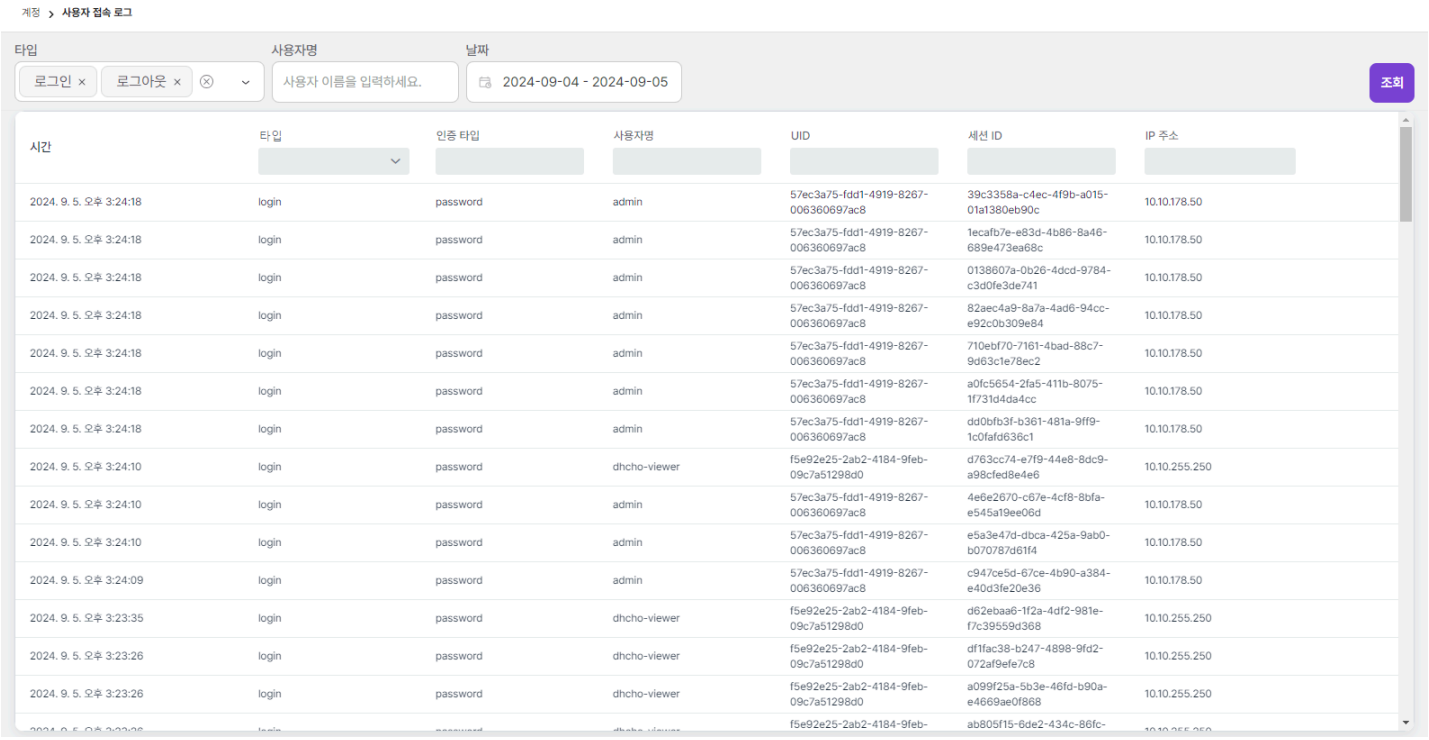

## 멤버 삭제

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.




| 항목 | 설명 |
|---|---|
| 시간 | 시간 |
| 타입 | 접속 타입 ( LOGIN 또는 LOGOUT ) |
| 인증 타입 | 승인 타입(인증 방식)으로 아코디언 웹 콘솔에서 로그인이 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 |

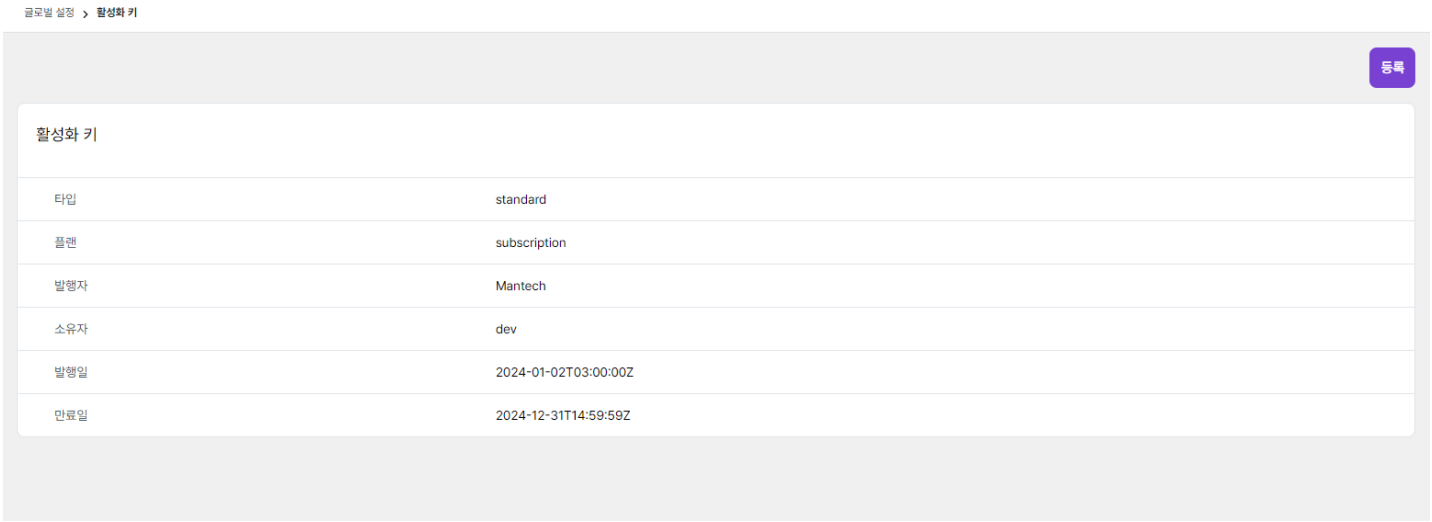
**TIP** 조회 후 테이블 헤더에 조건을 설정하면 결과 내 검색이 가능하다.

## 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

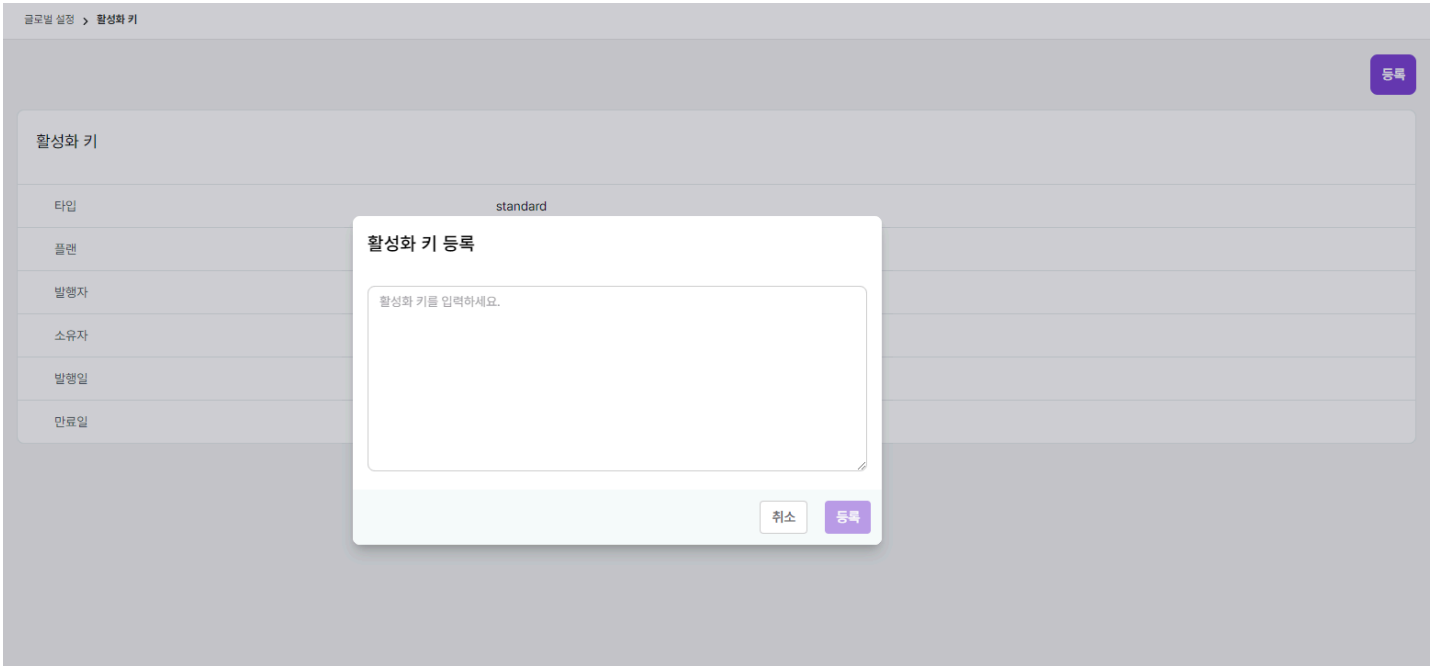
아코디언을 사용하기 위한 활성화 키를 관리한다.




| 분류 | 설명 |
| --- | --- |
| 타입 | Standard 또는 Trial<br>• Standard<br>◦ 정식 활성화 키에 의해 관리받을 수 있는 타입<br>• Trial<br>◦ 일정 기간동안 아코디언을 사용해 볼 수 있는 타입 |
| 플랜 | Perpetual 또는 Subscription<br>• Perpetual<br>◦ 영구적인 활성화 키<br>• Subscription<br>◦ 기간이 있는 활성화 키 |
| 발행자 | 활성화 키 발행인 (Mantech) |
| 소유자 | 활성화 키 소유자 |
| 발행일 | 활성화 키 발행일 |
| 만료일 | 활성화 키 만료일 |

## 활성화 키 등록

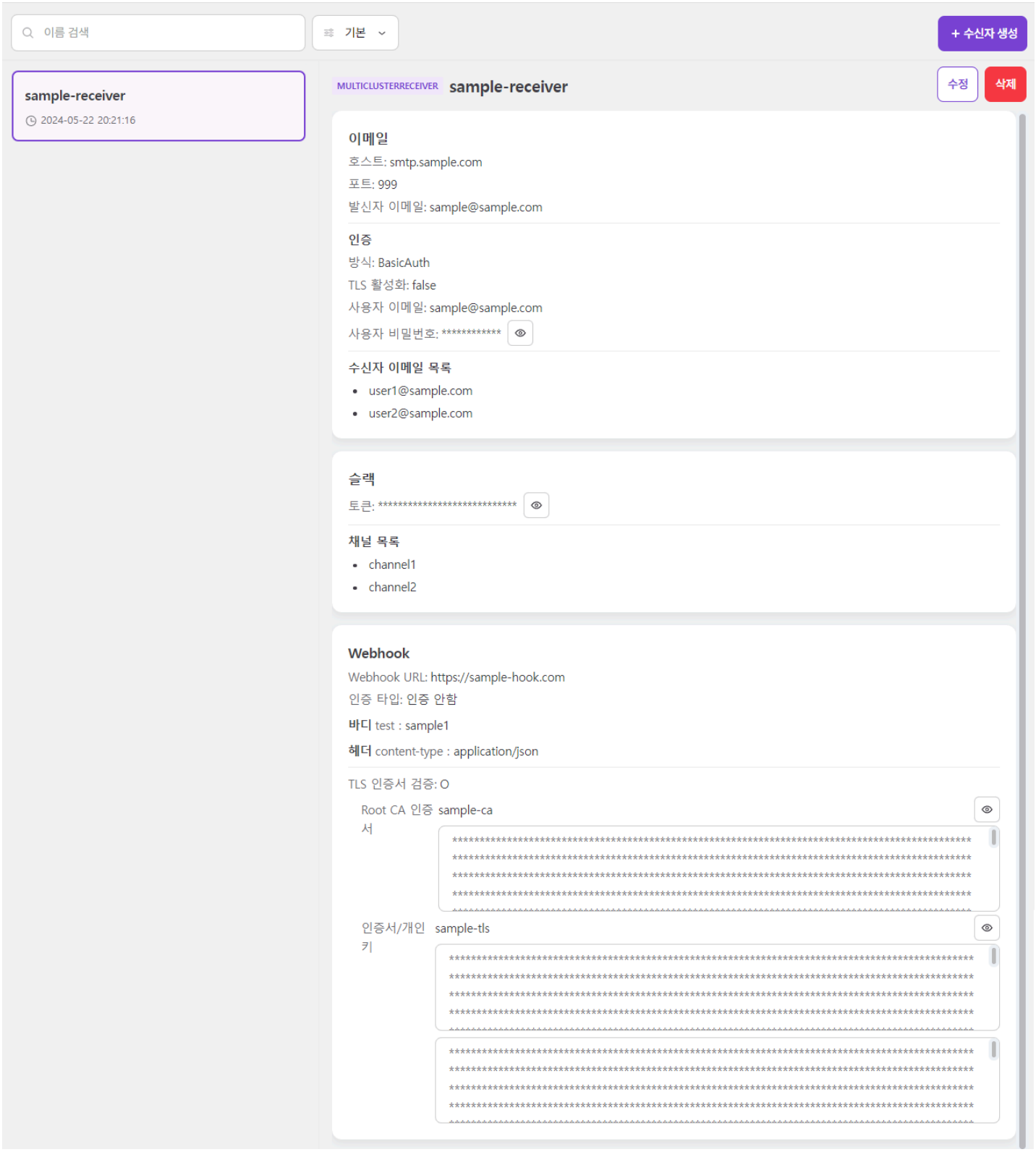
우측 상단의 등록 버튼을 선택한다. 모달이 나타나면 활성화 키를 입력하여 등록한다.



## 4.1.5.2. 글로벌 수신자

알림 정책에 따라 알림을 받을 수신자를 설정한다.

### 글로벌 수신자 개요



- 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 |

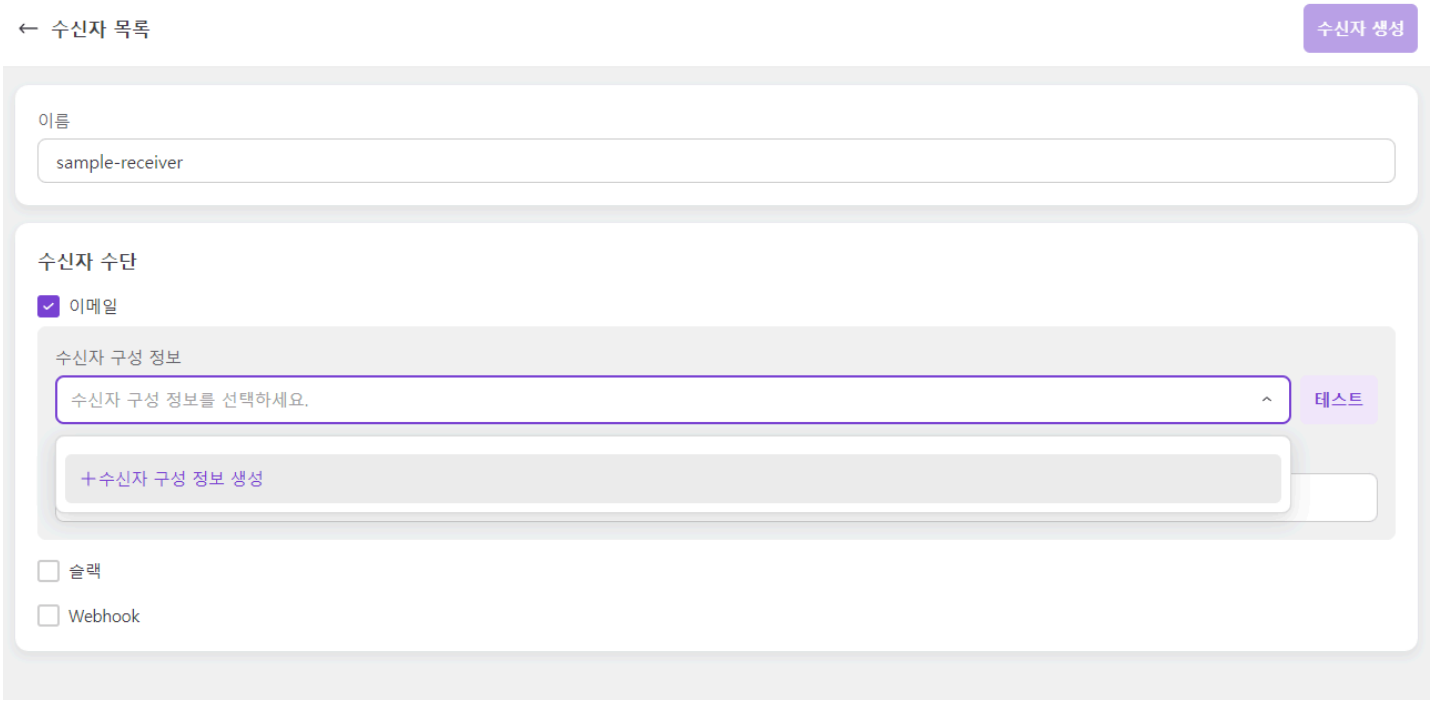
- 웹훅

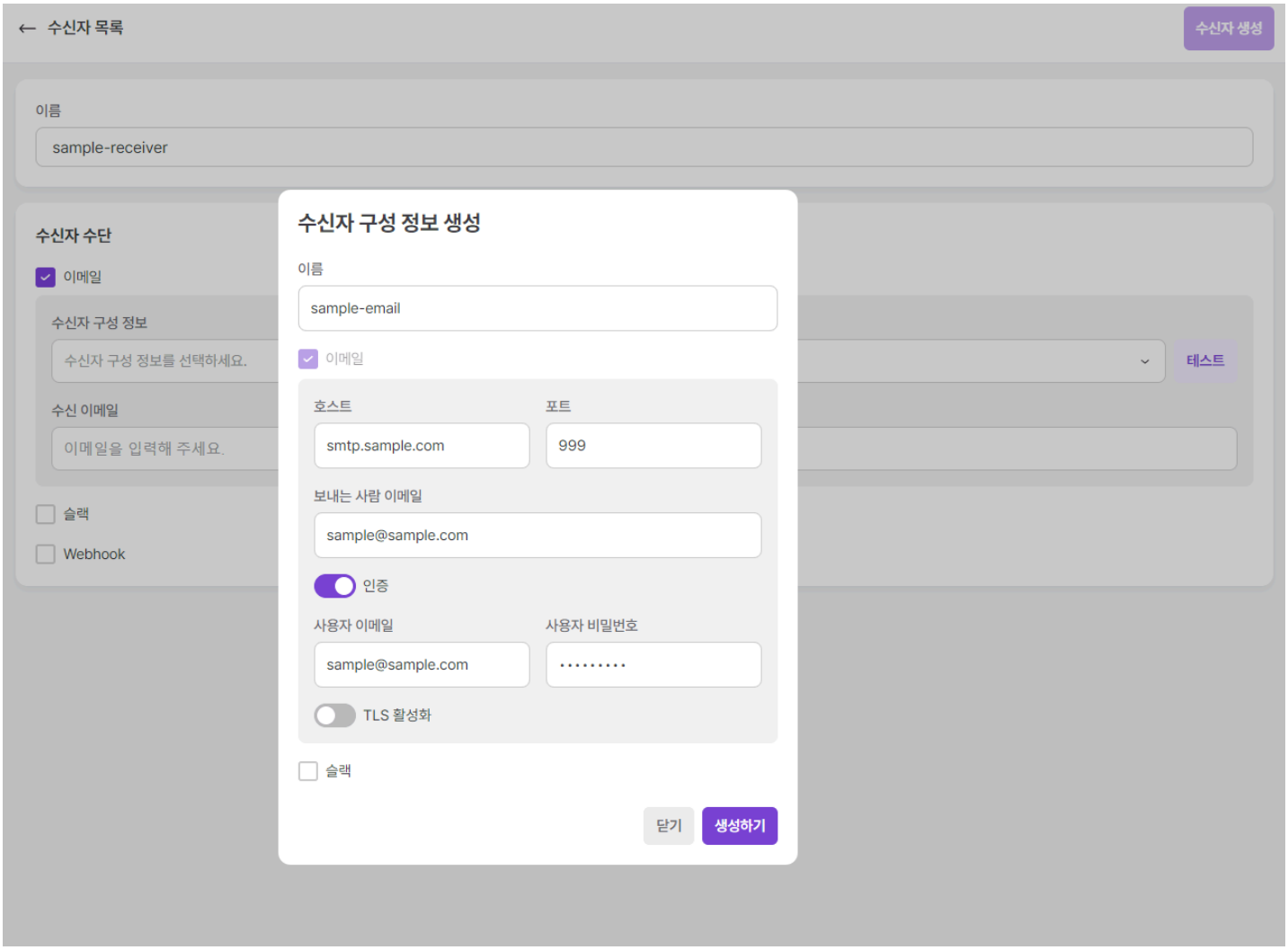
| 분류 | 설명 |
| --- | --- |
| Webhook URL | 웹훅 URL 주소 |
| 인증 타입 | `인증 안함` (익명), `Bearer 인증` , `basic 인증` |
| 토큰 | `Bearer 인증` 의 경우 토큰 정보 |
| 사용자 이름 | `basic 인증` 의 경우 사용자 이름 |
| 사용자 비밀번호 | `basic 인증` 의 경우 사용자 비밀번호 |
| 바디 | 요청에 포함될 바디 |
| 헤더 | 요청에 대한 헤더 |
| TLS 인증서 검증 | 검증 시 `O` , 미검증 시 `X` |
| Root CA 인증서 | 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**

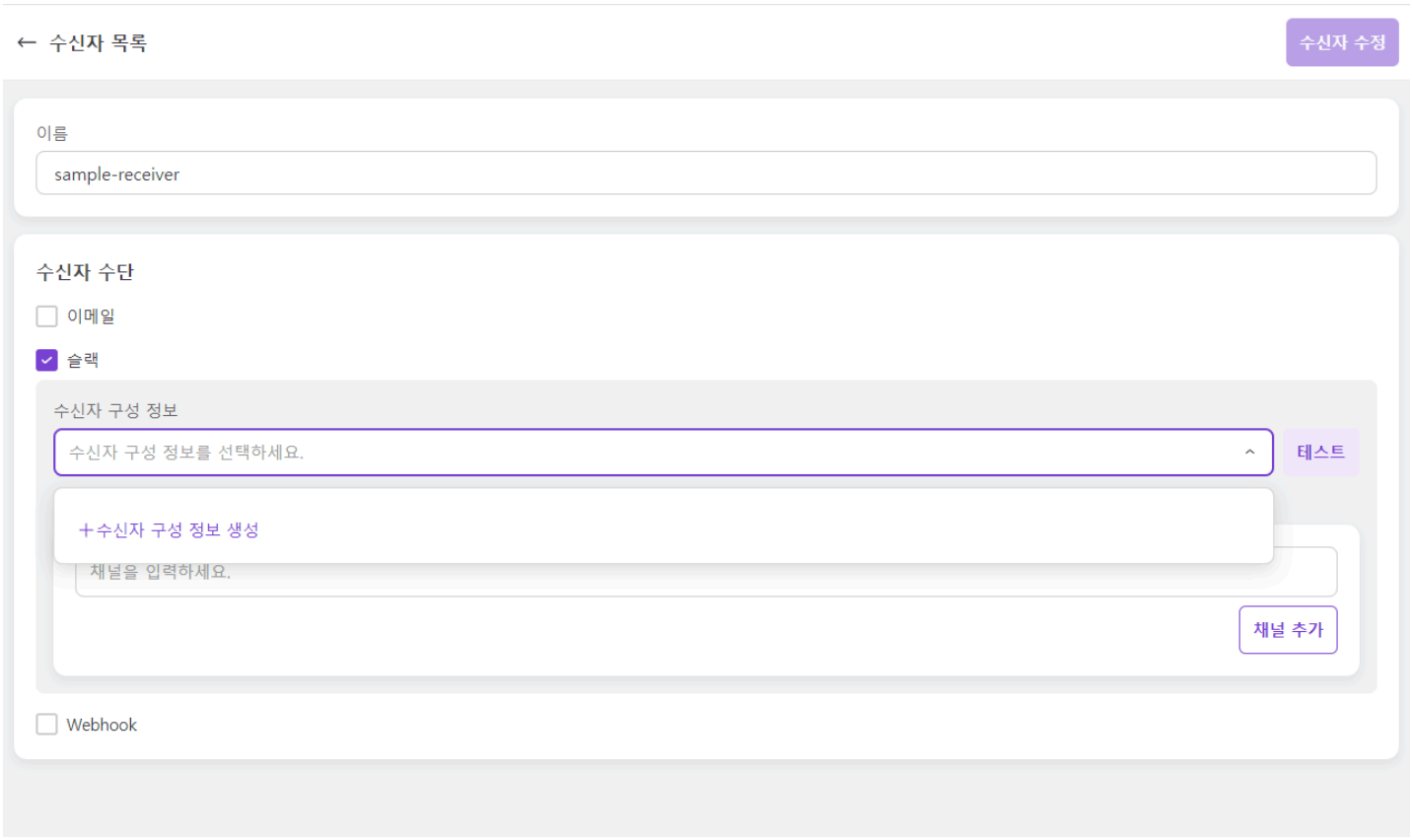
수신 이메일

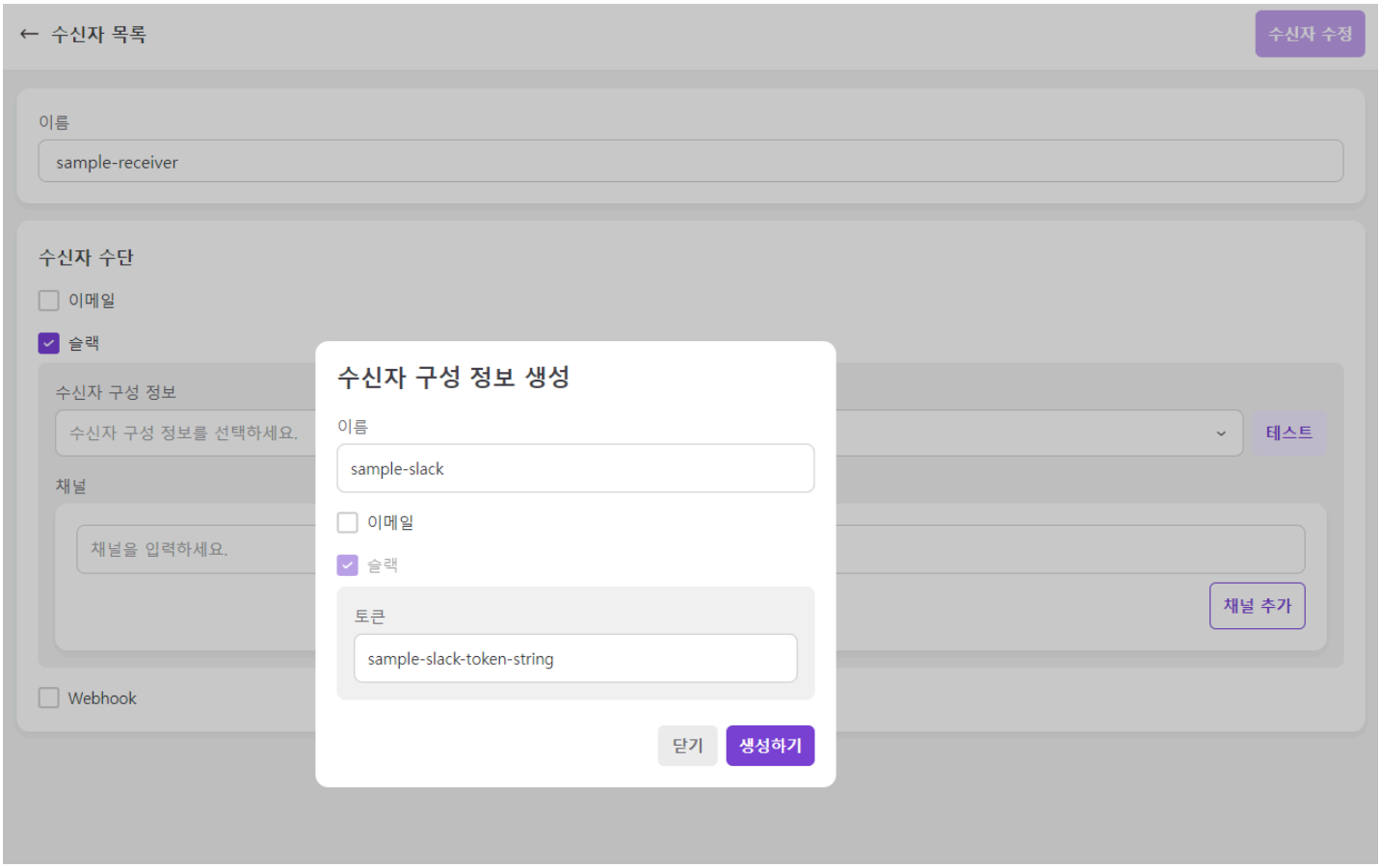
user1@sample.com × user2@sample.com × 이메일을 입력해 주세요.

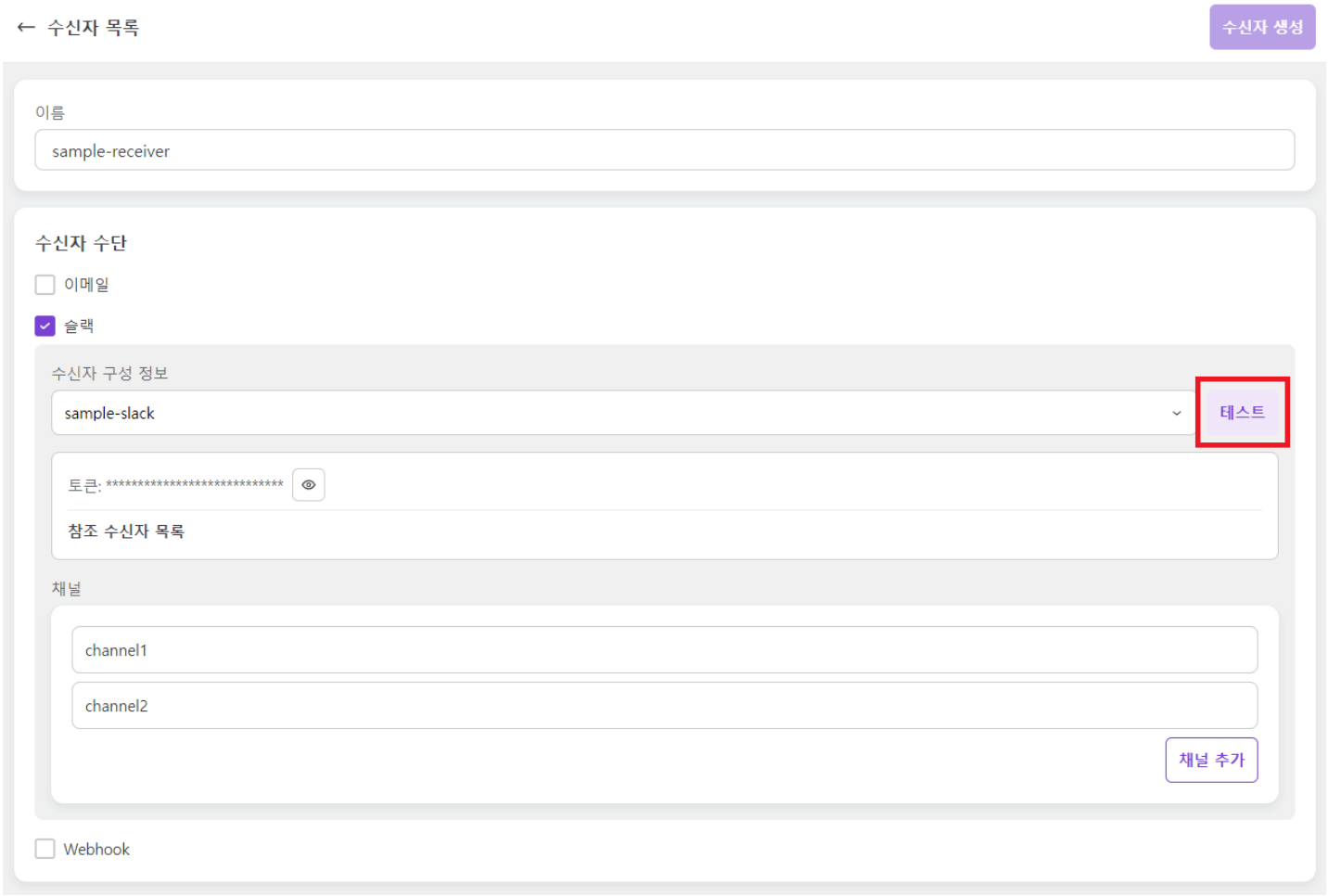
슬랙

Webhook

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



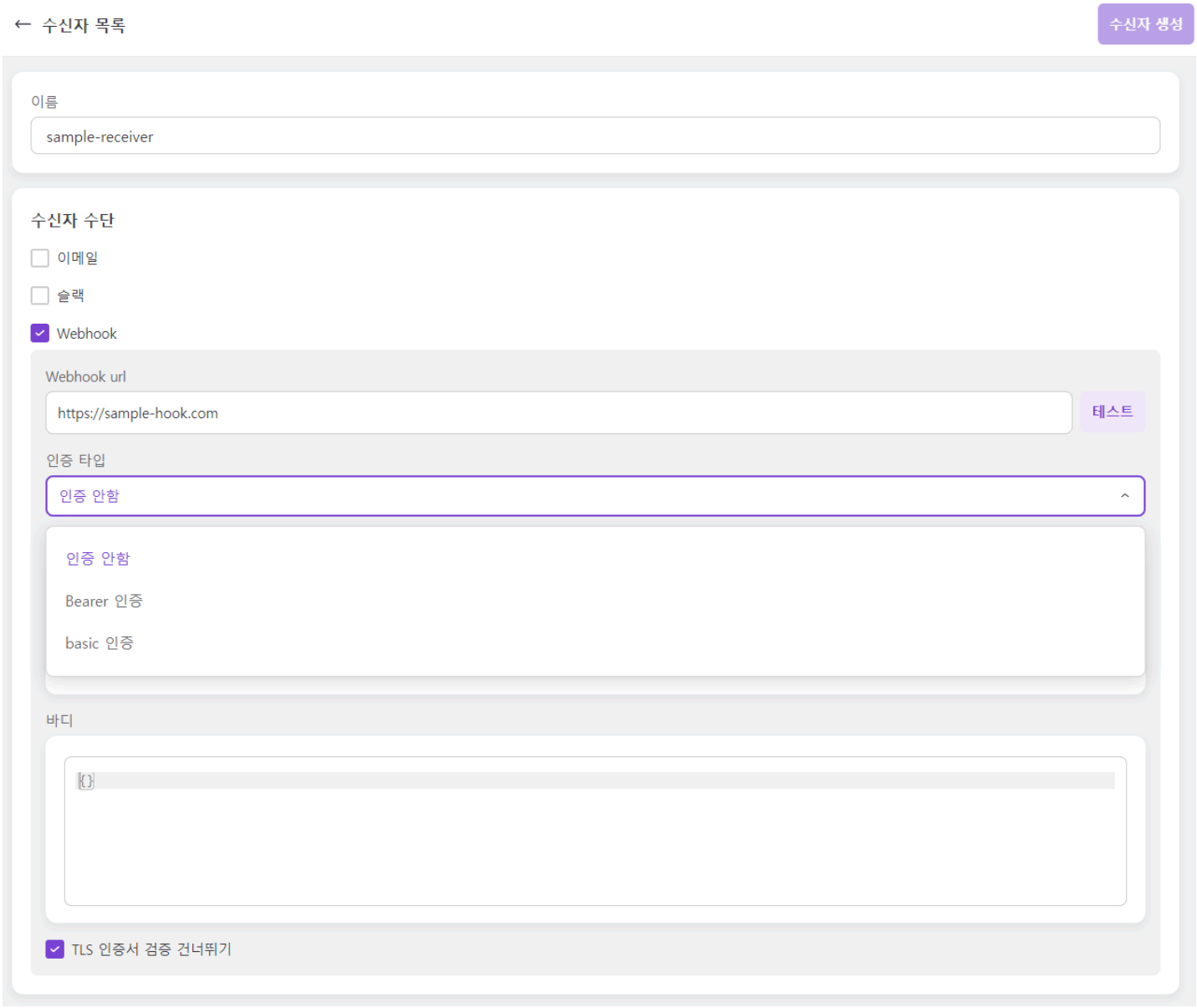

← 수신자 목록
수신자 생성
이름
sample-receiver
수신자 수단
이메일
슬랙
수신자 구성 정보
sample-slack
테스트
토큰: **********
참조 수신자 목록
채널
channel1
channel2
채널 추가
Webhook

웹훅 설정

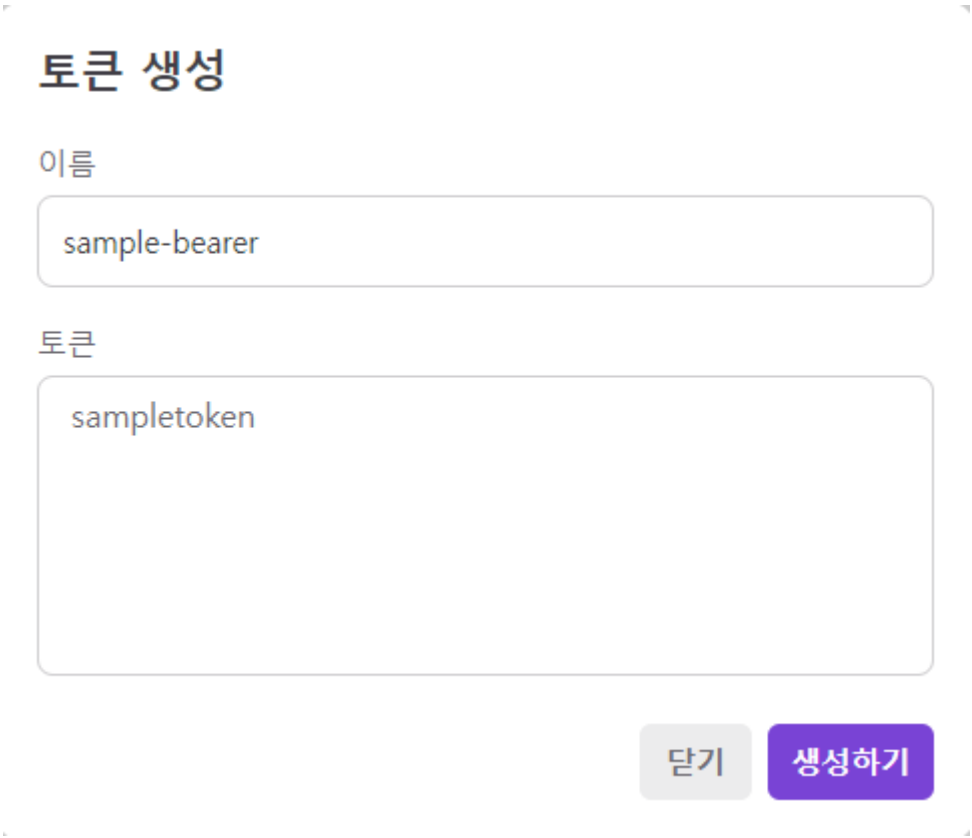
웹훅 URL, 헤더, 바디 정보를 입력한다.

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

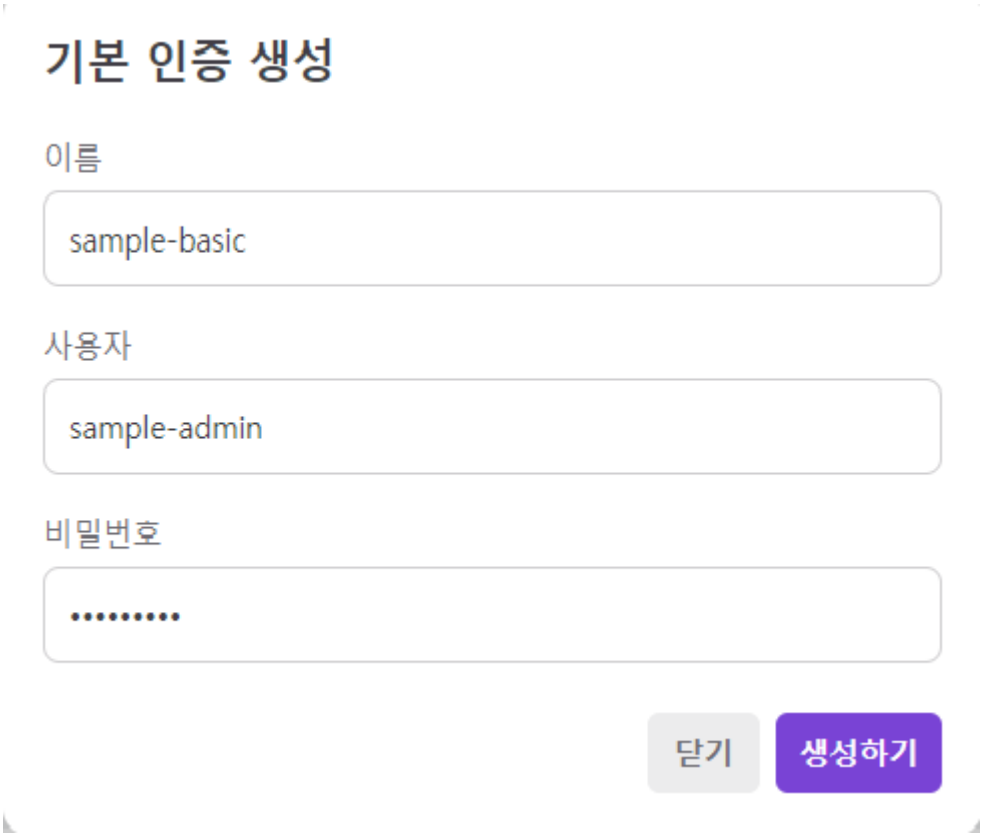
웹훅 URL, 헤더, 바디 정보 입력 및 인증 타입 선택 후 `테스트` 버튼을 눌러 유효한 설정인지 확인한다.



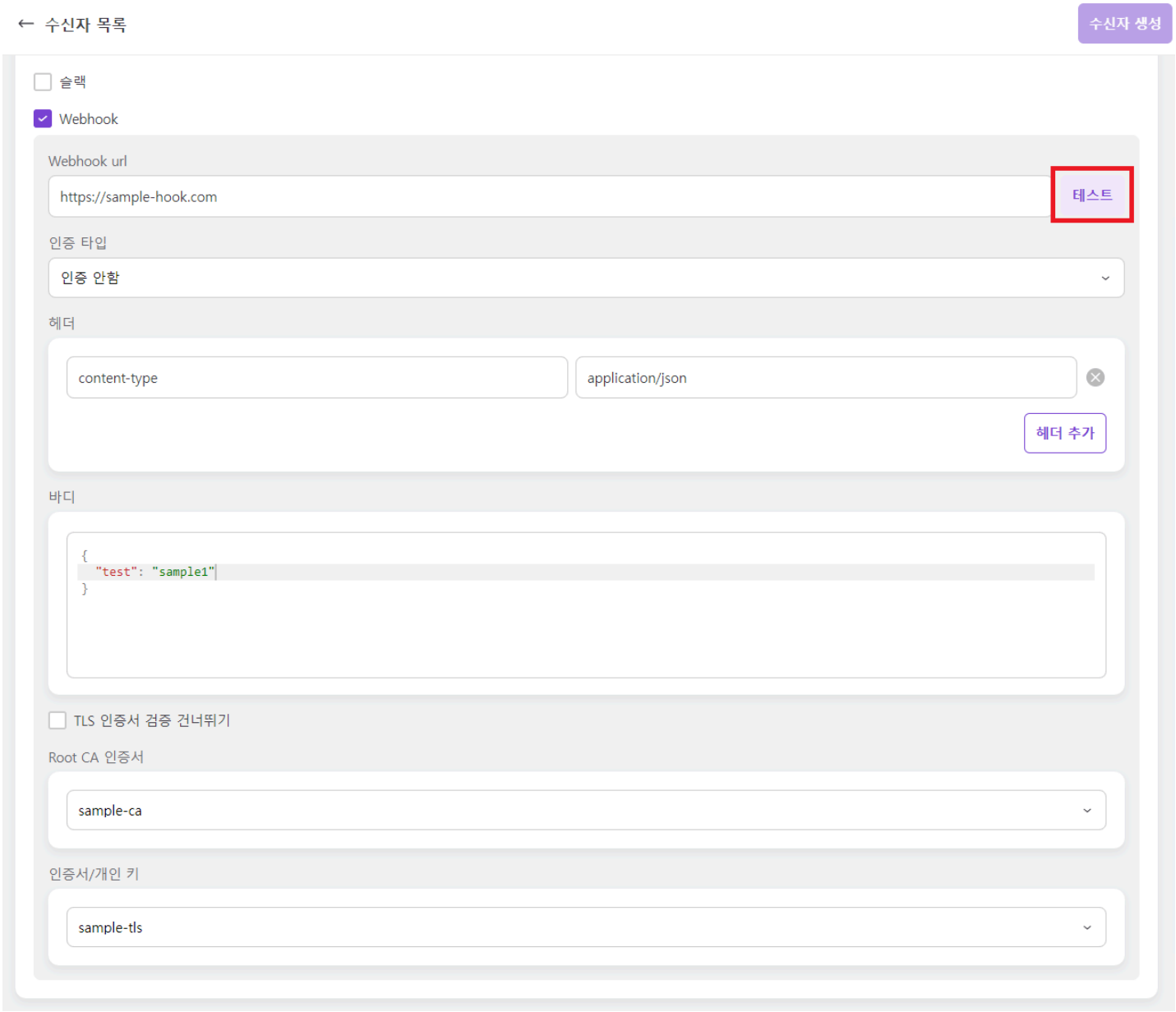
Bearer 인증 선택 시 토큰 정보가 담긴 시크릿 생성 및 선택




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




TLS 인증서 검증 건너뛰기 해제 시
Root CA 인증서 및 인증서/개인키를 동일한 방법으로 생성 및 선택




**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.

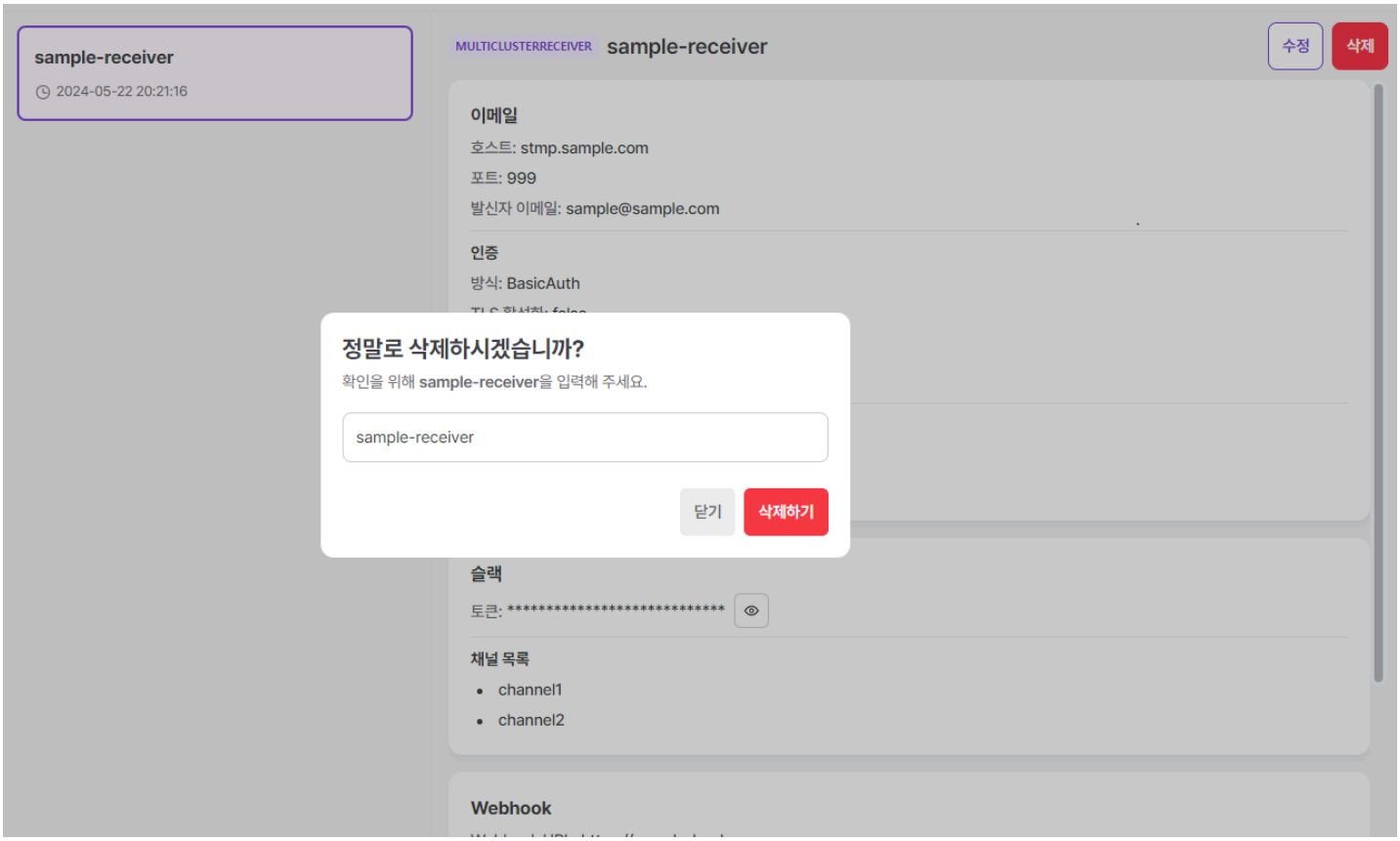
- 정보 유출 위험: 복사된 수신자/수신자 구성 정보는 모든 멤버 클러스터에 배포되므로, 각 클러스터의 보안 수준이 매우 중요합니다. 클러스터에 접근 권한을 철저히 관리하여 무단 접근을 방지하세요.
  클러스터 수신자/수신자 구성 정보 중 기밀정보는 acc-system 네임스페이스의 secret으로 관리됩니다. 클러스터 관리자의 권한 사용자는 기밀정보의 접근이 가능하나, 그 외 네임스페이스 접근 사용자는 조회를 할 수 없습니다.
  자세한 내용은 Kubernetes의 RBAC(역할 기반 접근 제어) (https://kubernetes.io/docs/reference/access-authn-authz/rbac/)를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

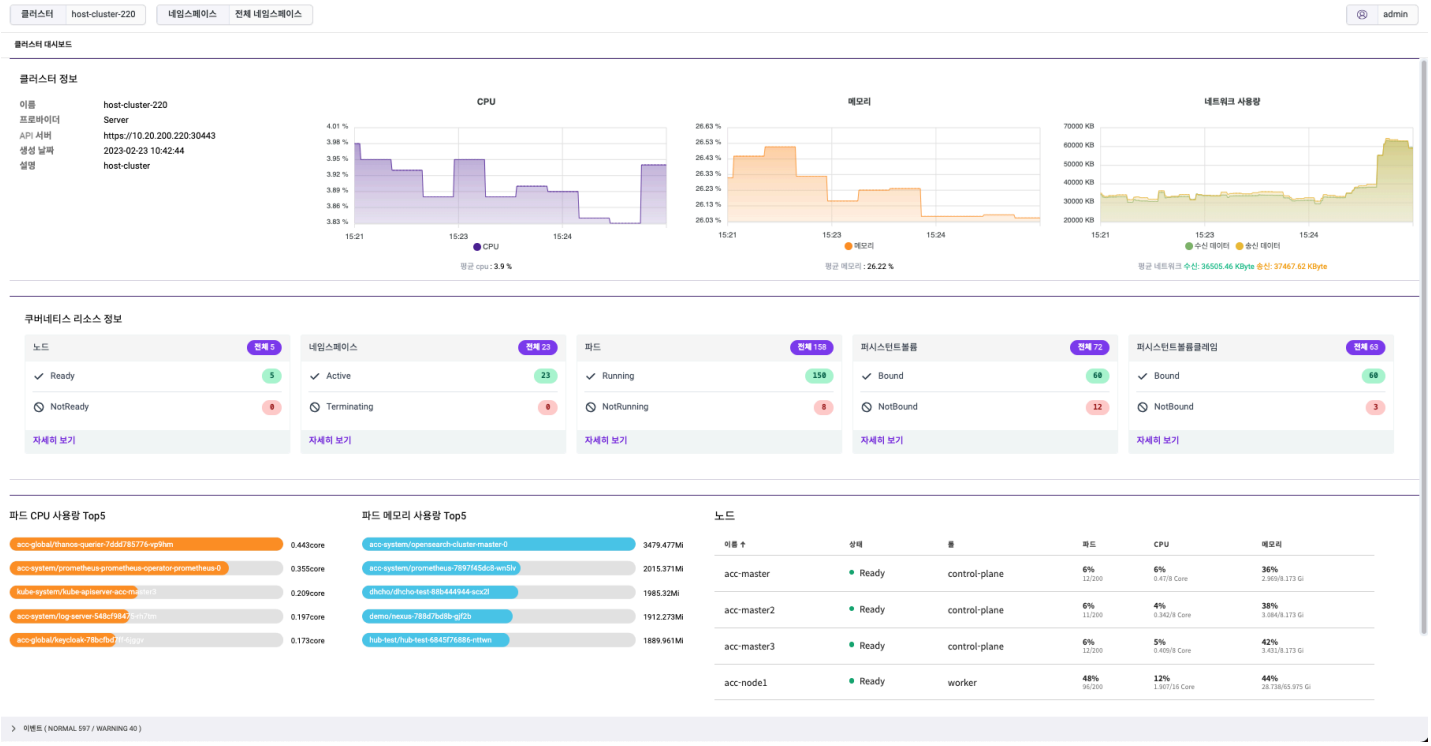



모달에서 수신자 이름을 입력 후 `삭제하기` 버튼을 선택한다.

# 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 | 클러스터 기본 정보 및 CPU, 메모리, 네트워크 사용 정보를 제공한다.<br>• 이름: 클러스터 이름<br>• 프로바이더: 쿠버네티스 제공자<br>• API 서버: 아코디언 멤버 에이전트 엔드포인트<br>• 생성 날짜: 클러스터 등록 시간<br>• 설명: 클러스터 설명<br>**NOTE** 생성 날짜 시간 기준은 UTC 이다. |
| 쿠버네티스 리소스 정보 | 클러스터에 배포된 쿠버네티스 리소스 정보를 제공한다. 자세히 보기 클릭 시 해당 리소스의 목록 화면으로 이동하며 상세 정보를 확인할 수 있다.<br>**NOTE** 파드의 경우 상태 를 클릭하면 해당 상태 로 필터링 된 파드 목록 화면으로 이동한다. |

| 항목 | 설명 |
| --- | --- |
| 파드 CPU/메모리 사용량 Top5 | 클러스터에 배포된 파드 중 CPU/메모리 사용량이 많은 순서대로 5개의 파드 정보를 제공한다. |
| 노드 | 클러스터를 구성하는 노드들의 정보를 제공한다.<br>• 이름: 노드 이름<br>• 상태: 노드 상태 (Ready 또는 NotReady)<br>• 롤: 노드 역할 (control-plane 또는 worker)<br>• 파드: 노드에 배포된 파드 현황 (배포된 파드의 수/최대 배포가능한 파드의 수)<br>• CPU: 노드 CPU 사용 현황 (사용중인 코어 수/노드의 코어 수)<br>• 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.

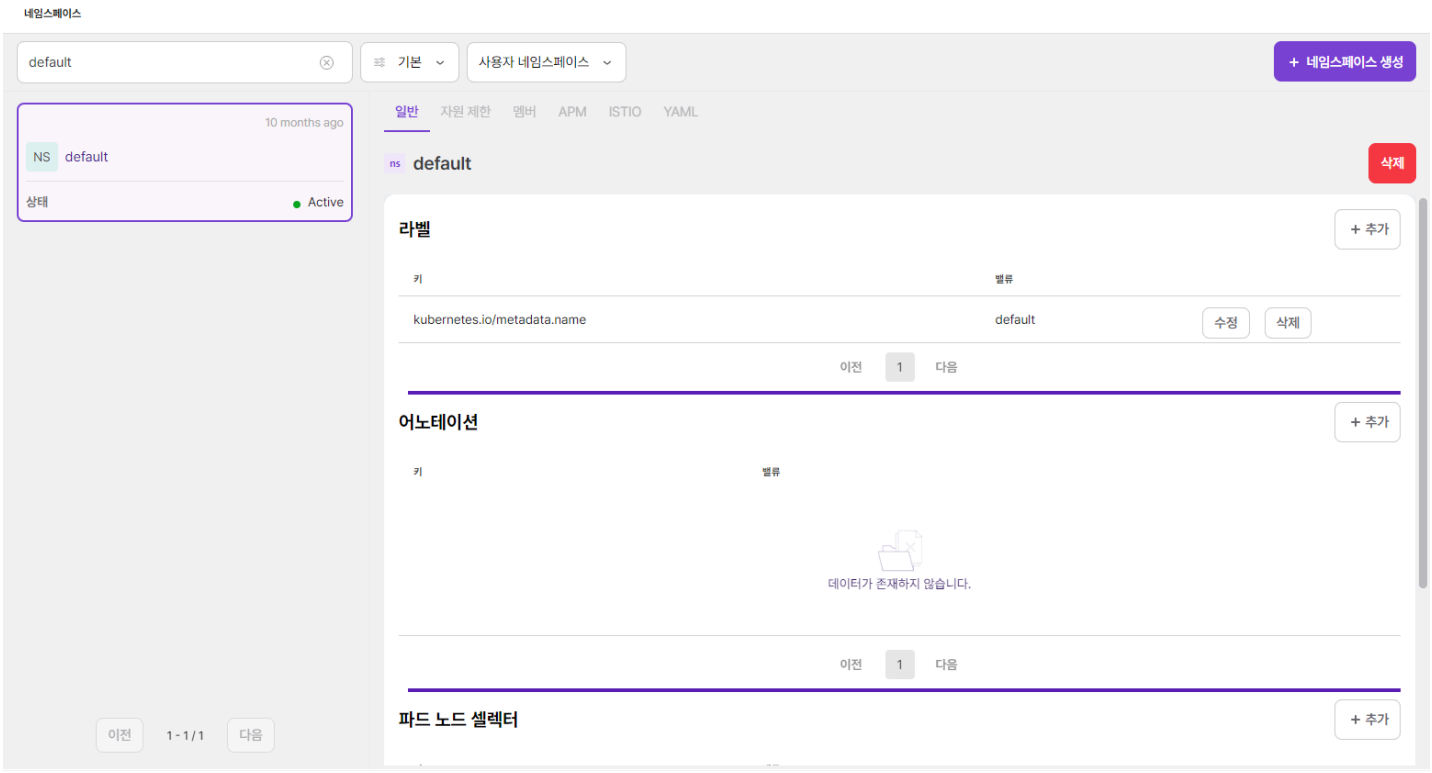



Table 1. 네임스페이스 목록

| 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 |

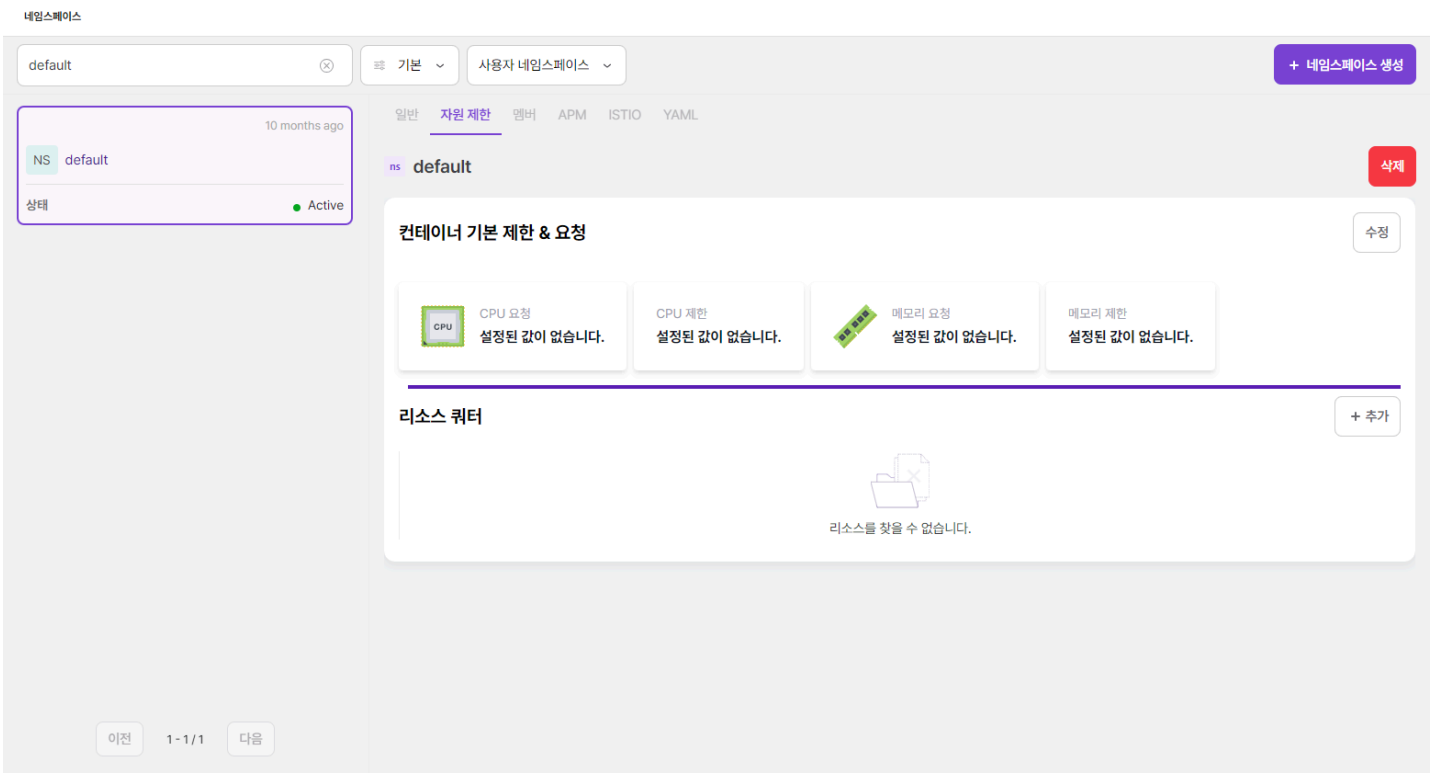



Table 3. 자원 제한

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

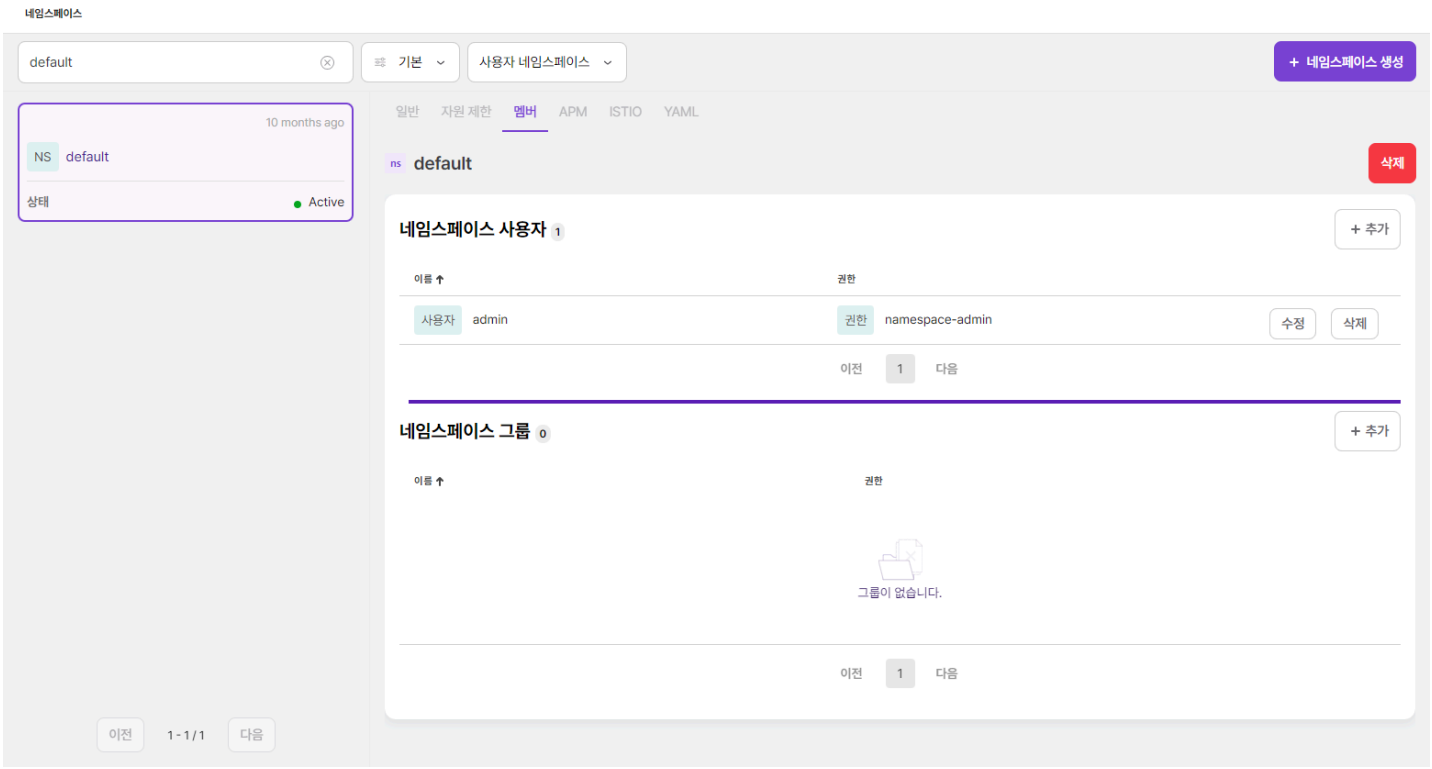



Table 4. 멤버

| 항목 | 설명 |
| --- | --- |

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

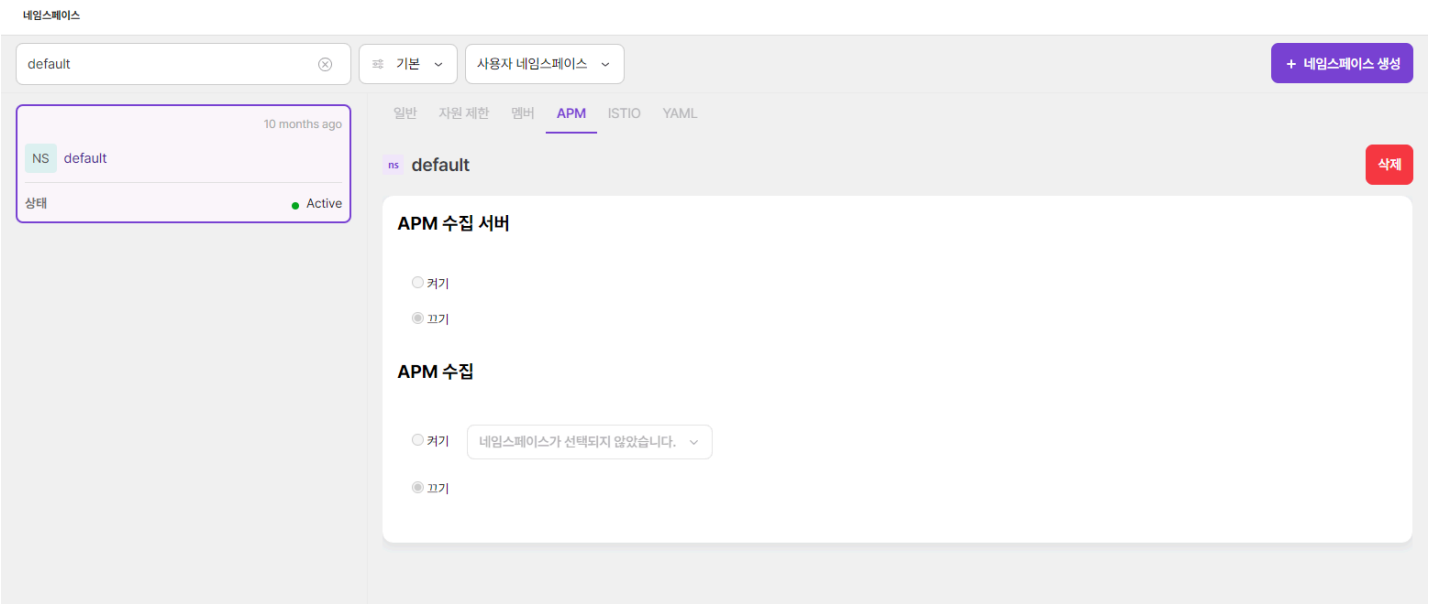



Table 5. APM

| 항목 | 설명 |
| --- | --- |
| APM 수집 서버 | 네임스페이스 내 스카우터 서버 배포 설정 |
| APM 수집 | 네임스페이스 내 스카우터 에이전트의 타겟 수집 서버 설정<br>(타겟 수집 서버의 네임스페이스를 선택) |

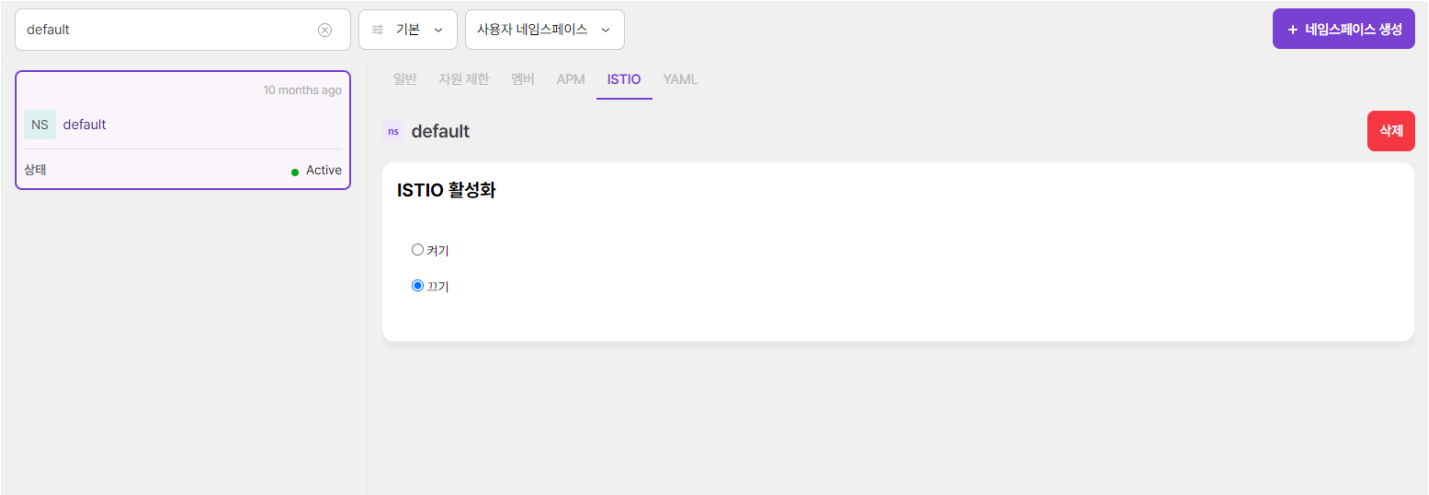



Table 6. ISTIO

| 항목 | 설명 |
| --- | --- |
| ISTIO 활성화 | istio-injection 수행 설정 |

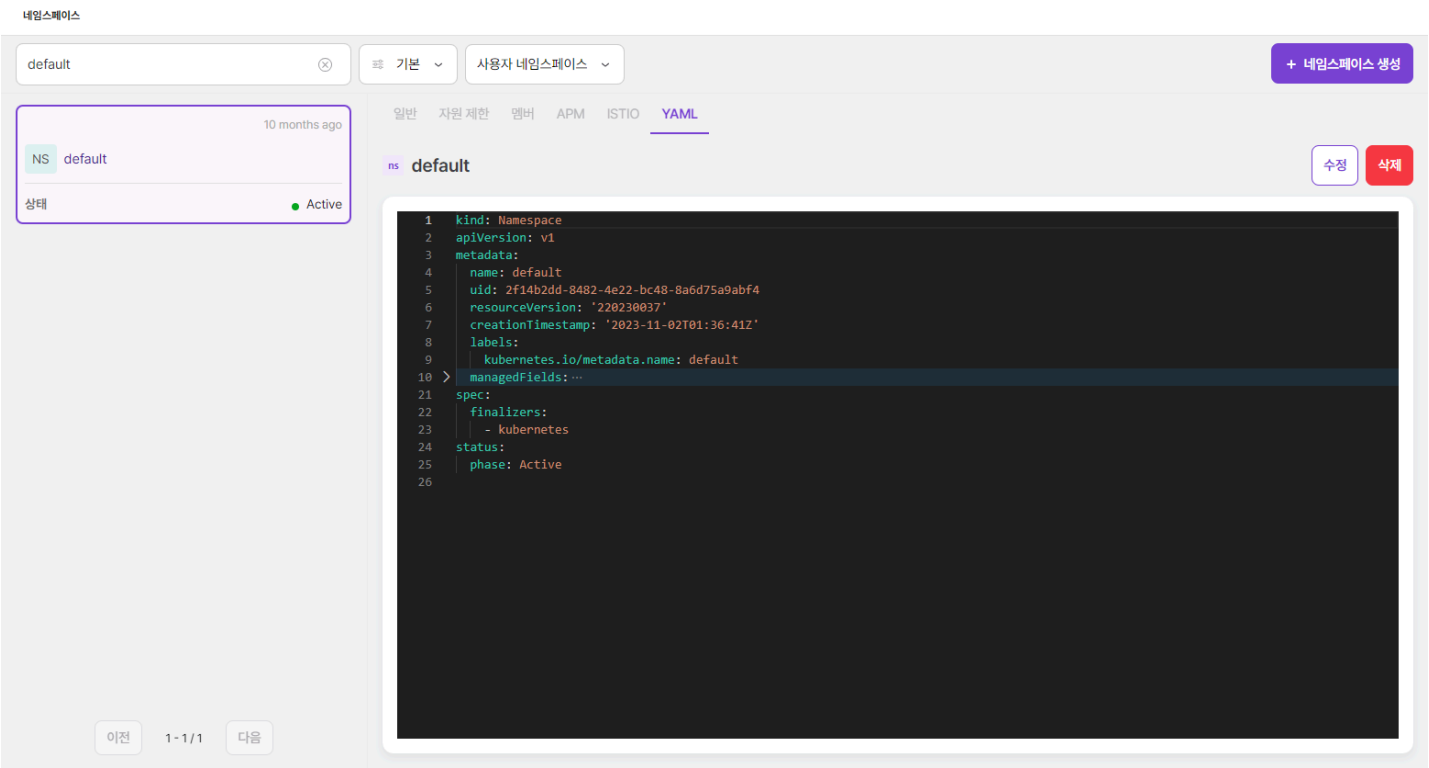
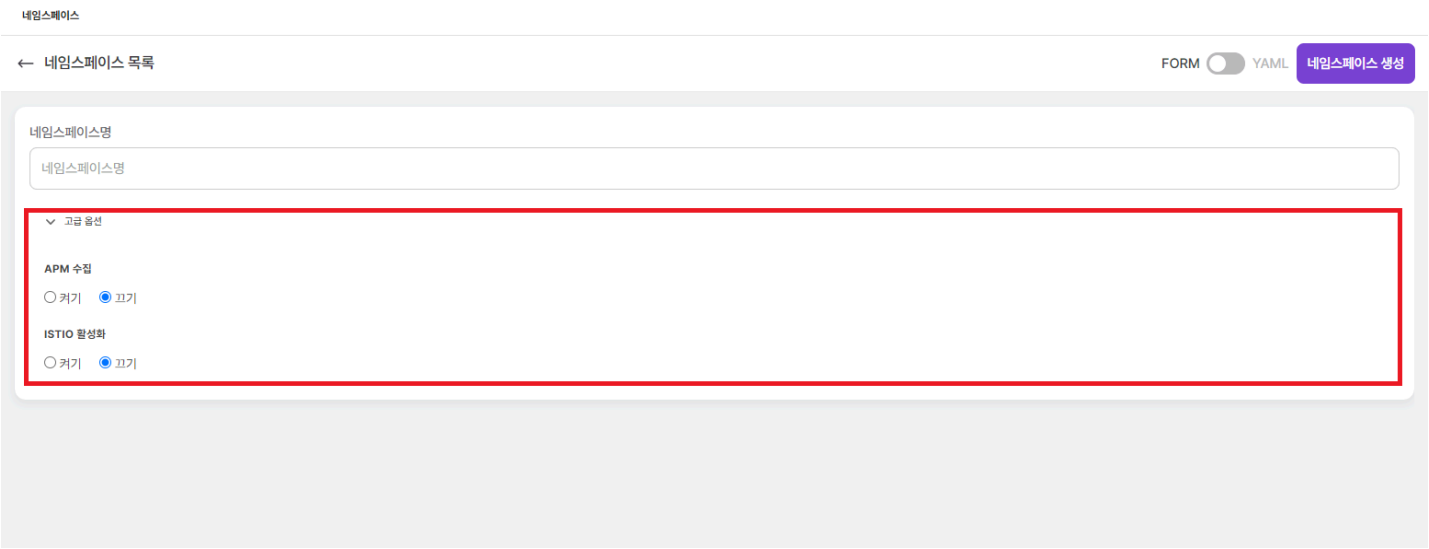



Table 7. YAML

| 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<br>(타겟 수집 서버의 네임스페이스를 선택) |
| ISTIO 활성화 | istio-injection 활성/비활성 |

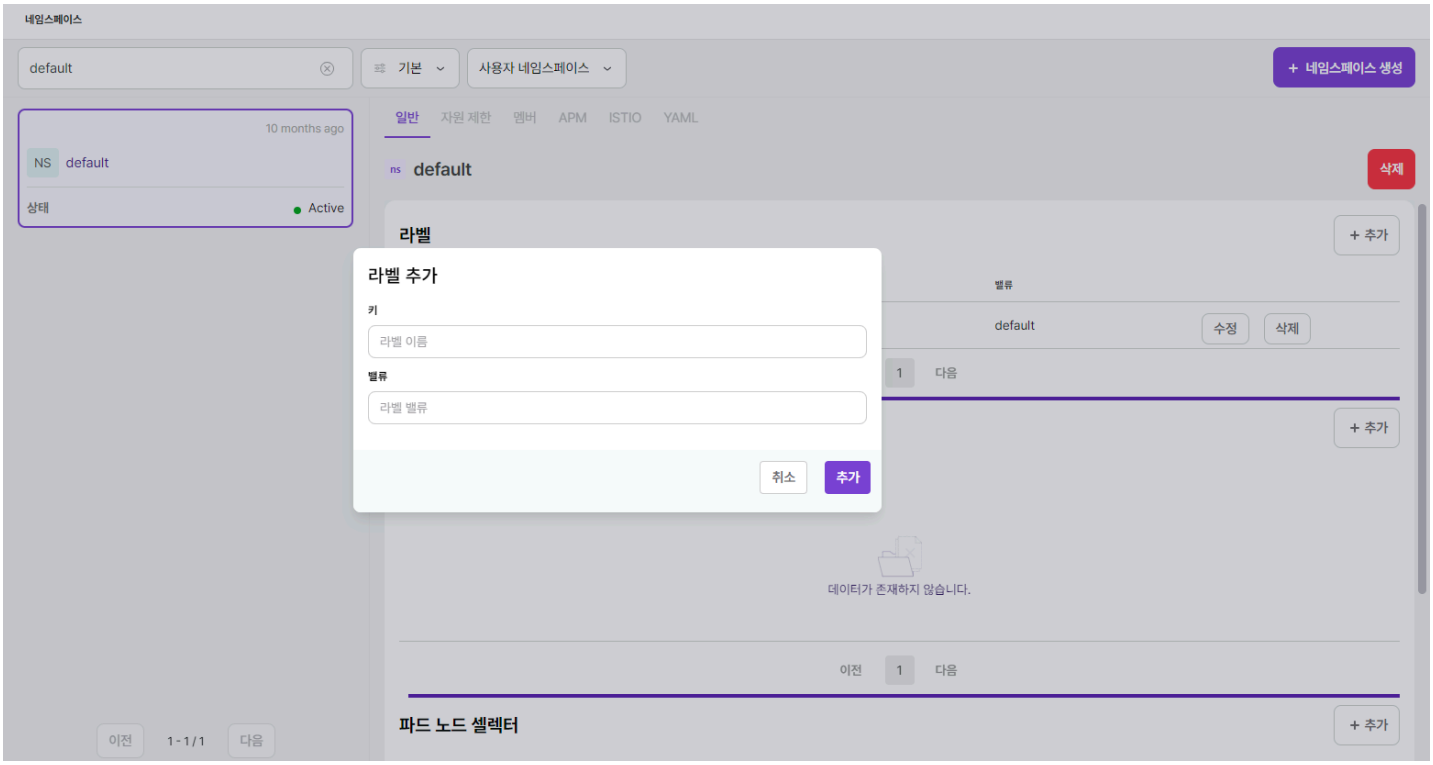
**NOTE**

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

## 4.2.2.2. 네임스페이스 수정

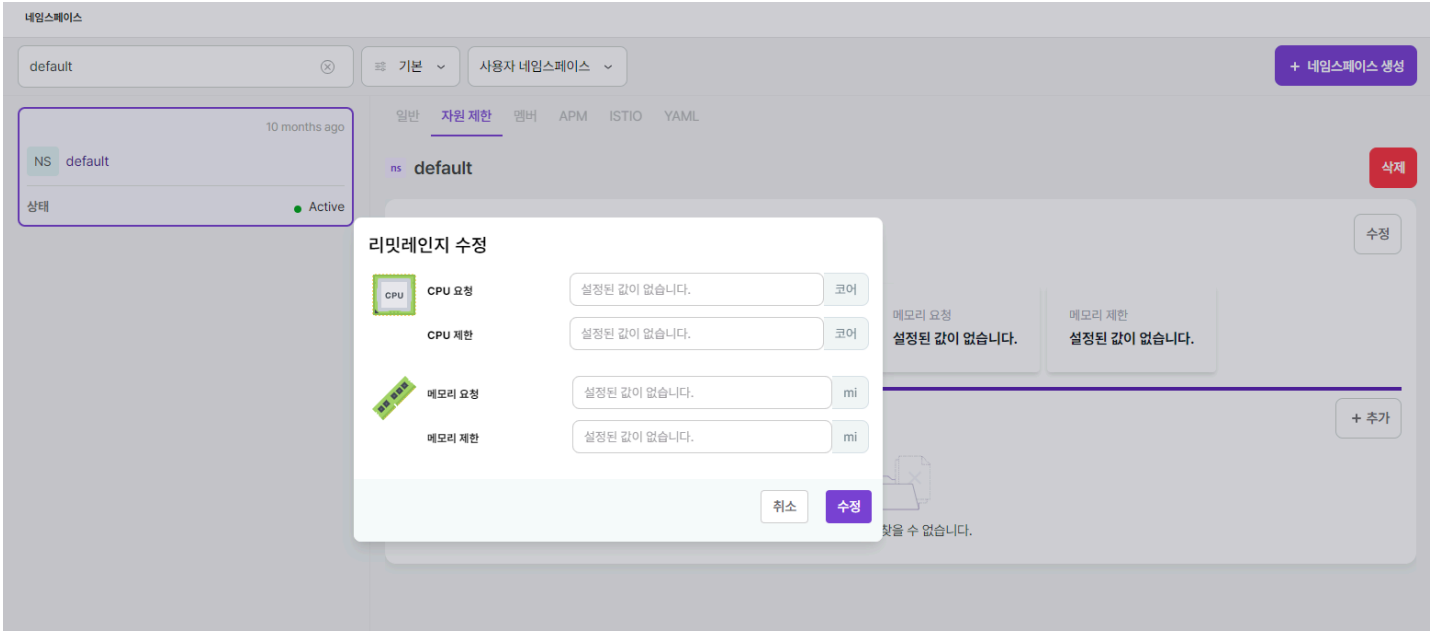
### 일반 정보 설정

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.



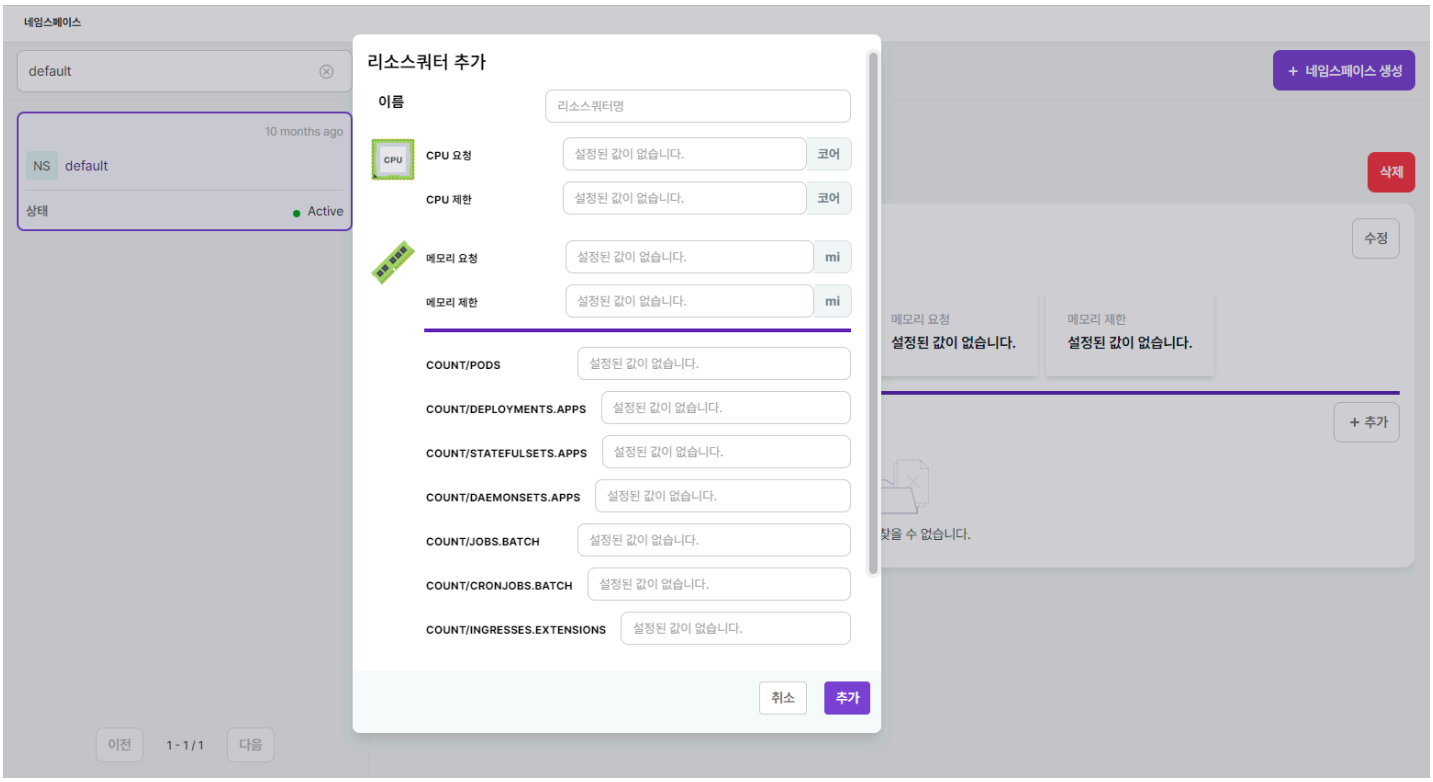

### 자원 제한 설정

쿠버네티스의 리밋레인지와 리소스쿼터를 이용해 네임스페이스 별로 자원을 제한할 수 있다.




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

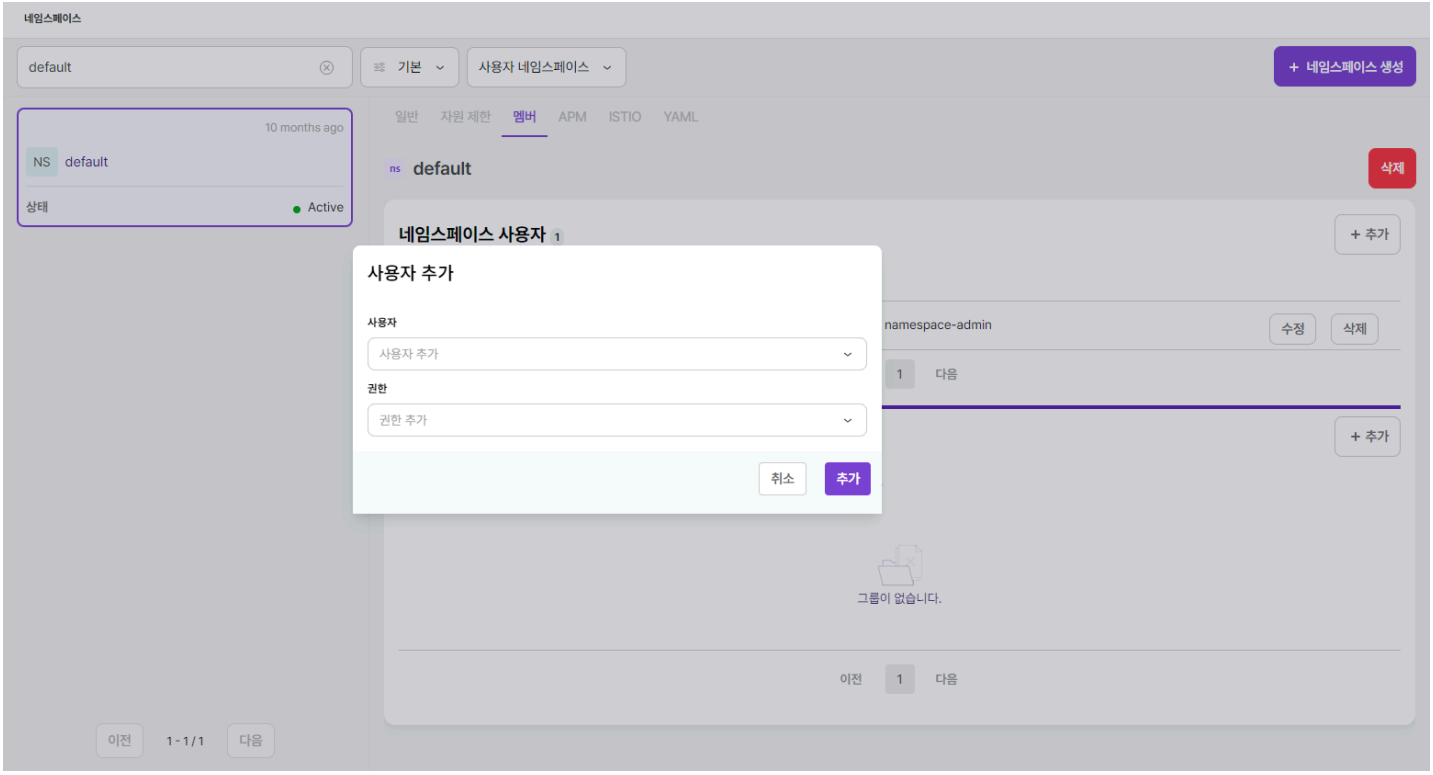
| 항목 | 설명 |
| --- | --- |
| 메모리 요청 | 컨테이너별 최소 메모리 요구량 |
| 메모리 제한 | 컨테이너별 최대 메모리 사용량 |




| 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.

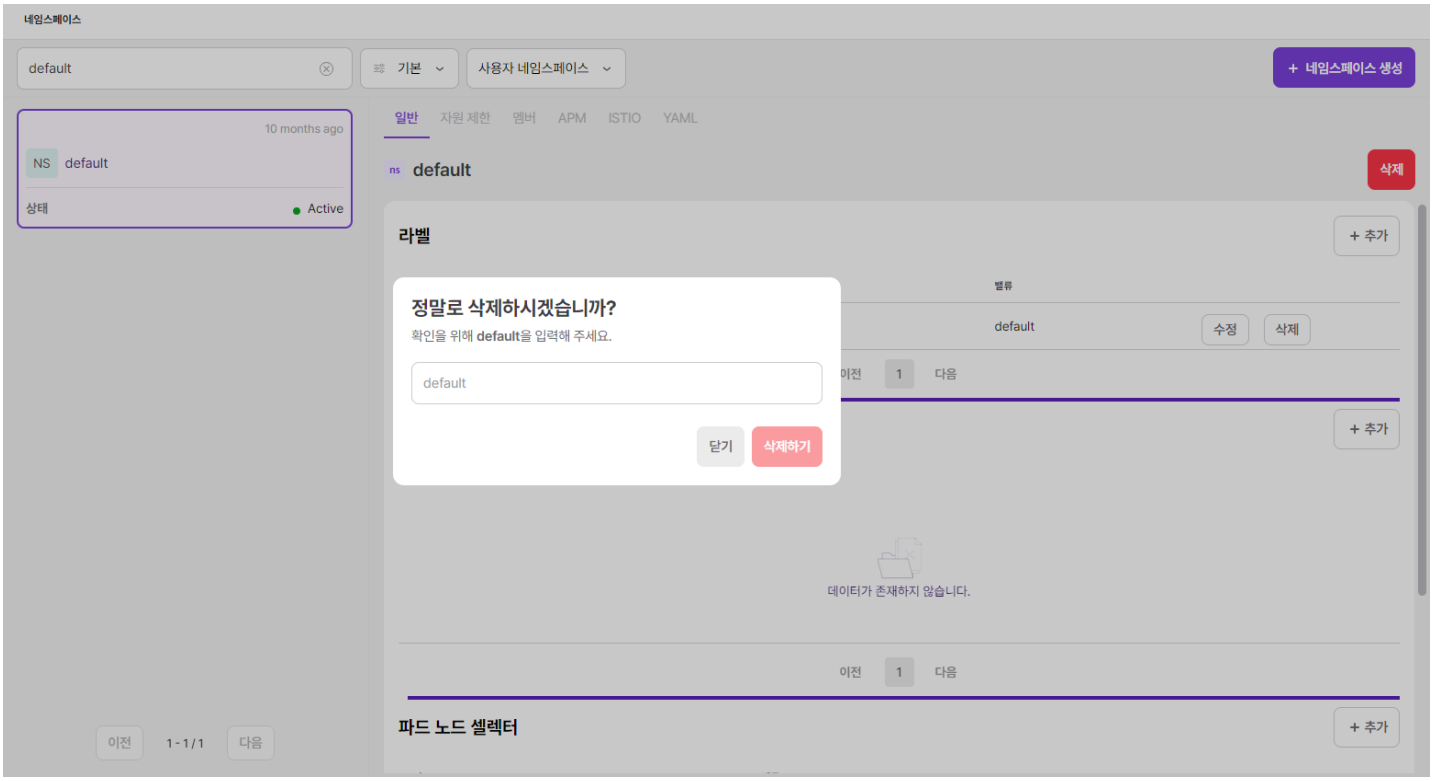



### APM 설정

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

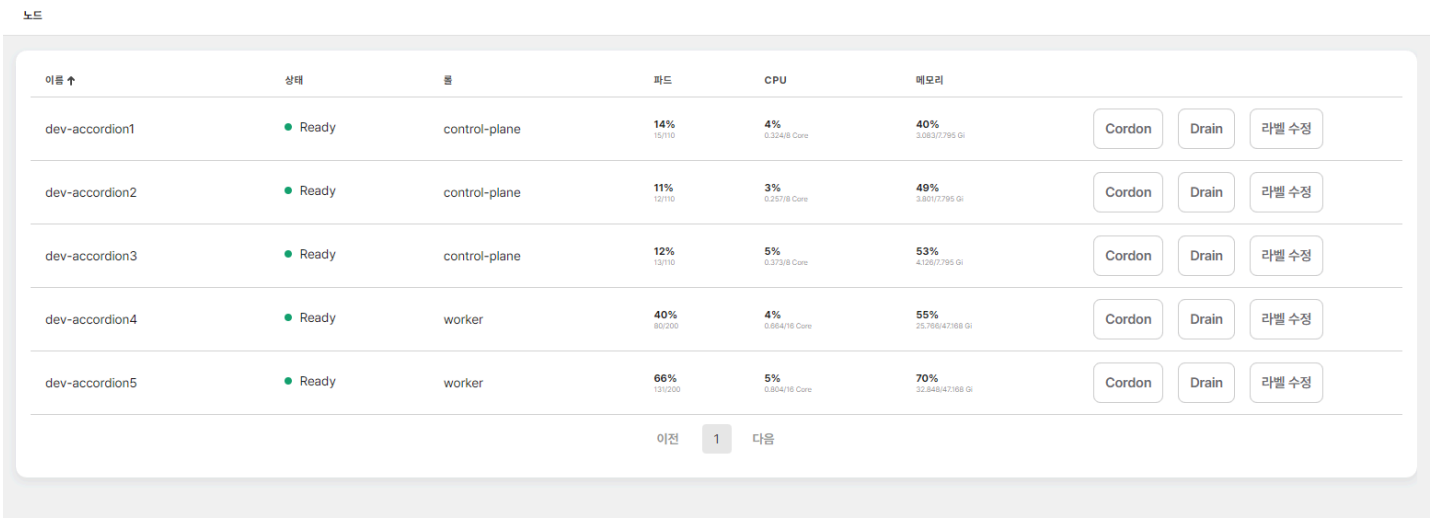
## 4.2.2.3. 네임스페이스 삭제

우측 상단의 삭제 버튼을 선택하여 해당 네임스페이스를 삭제할 수 있다.



### 4.2.3. 노드

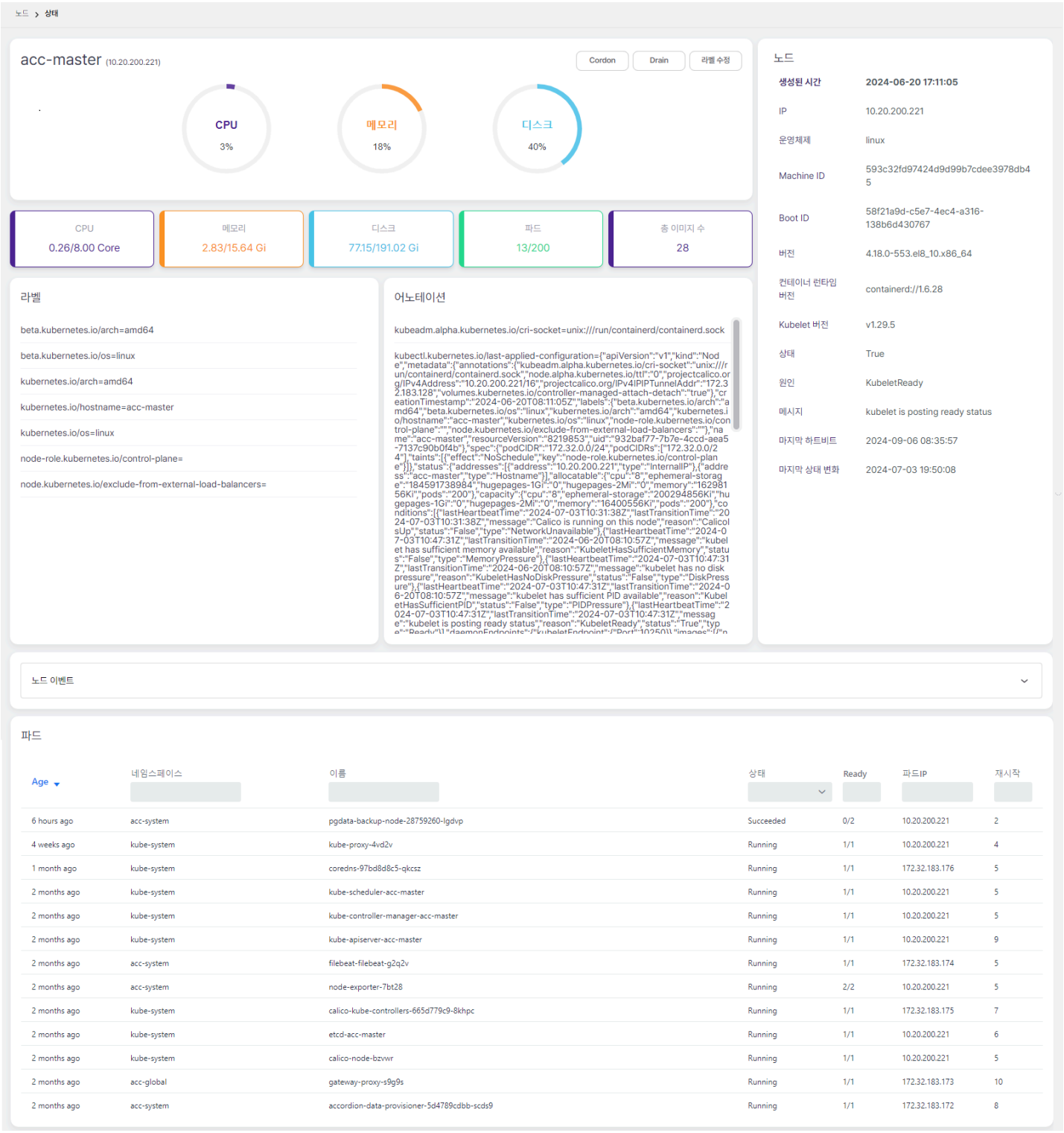
클러스터는 한개 이상의 노드로 구성되며 노드는 물리적 또는 가상 머신을 의미한다. 쿠버네티스는 파드를 노드에서 실행한다. 각 노드는 컨트롤 플레인에 의해 관리되며 파드 실행에 필요한 서비스를 포함한다. 노드는 역할에 따라 마스터(또는 컨트롤 플레인) 노드와 워커 노드로 구분된다.




노드 메뉴에서 제공하는 정보는 다음과 같다.

| 항목 | 설명 |
|---|---|
| 이름 | 노드 이름 |
| 상태 | 노드 상태 (정상일 경우 Ready 로 표시) |
| 롤 | 노드 역할 ( `control-plain` 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.

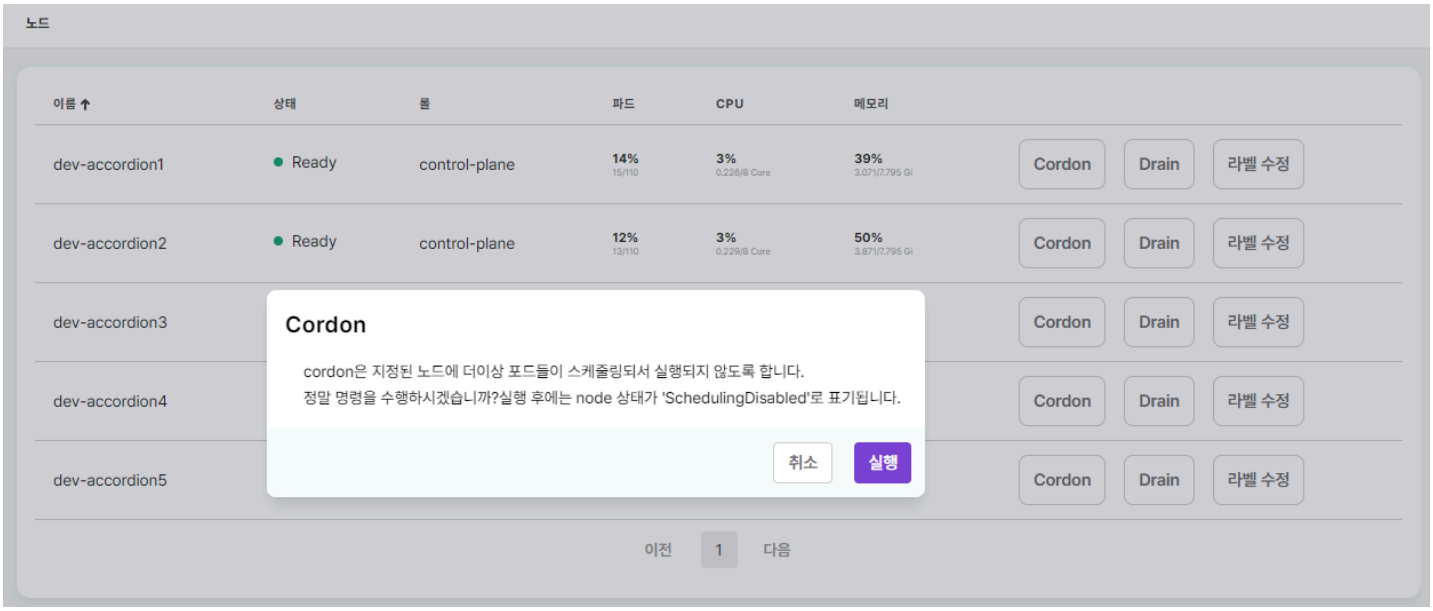



노드 메뉴(또는 노드 상세 화면)에서는 각 노드에 대해 `Cordon`, `Drain` 명령이 가능하고 노드의 레이블을 변경할 수 있다.

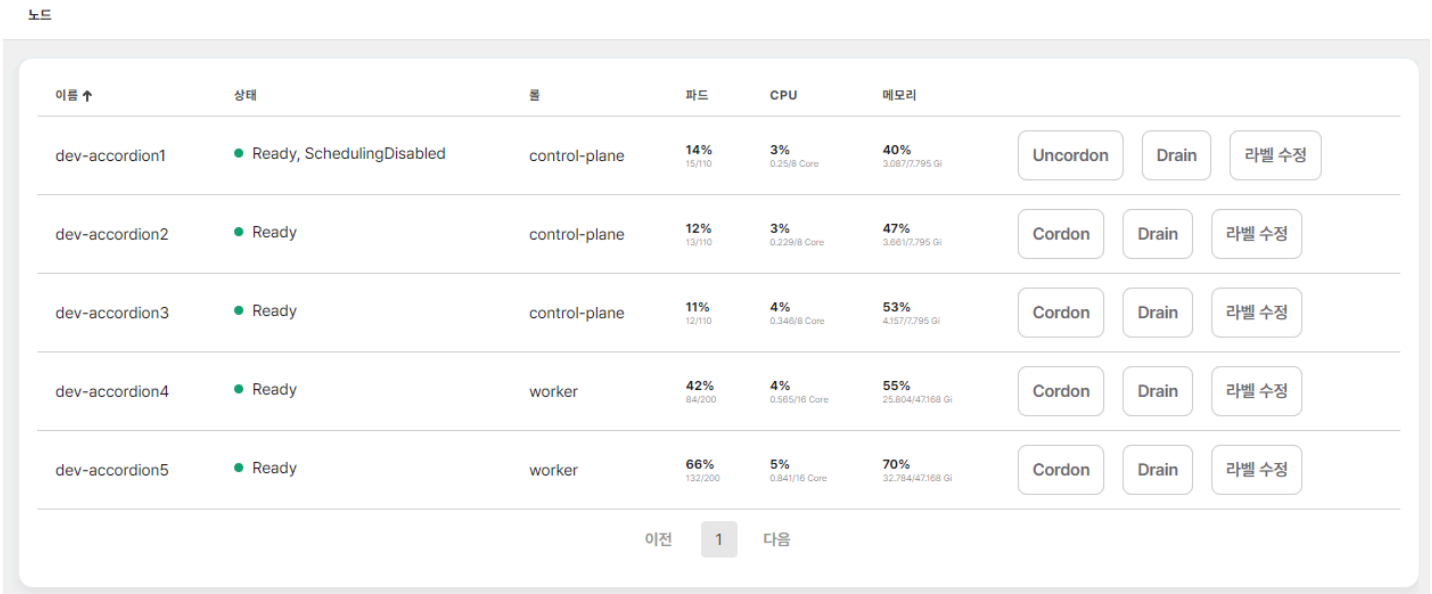
**TIP** 문서에서는 노드 메뉴에서 설정하는 방법을 기준으로 가이드한다.

### 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.



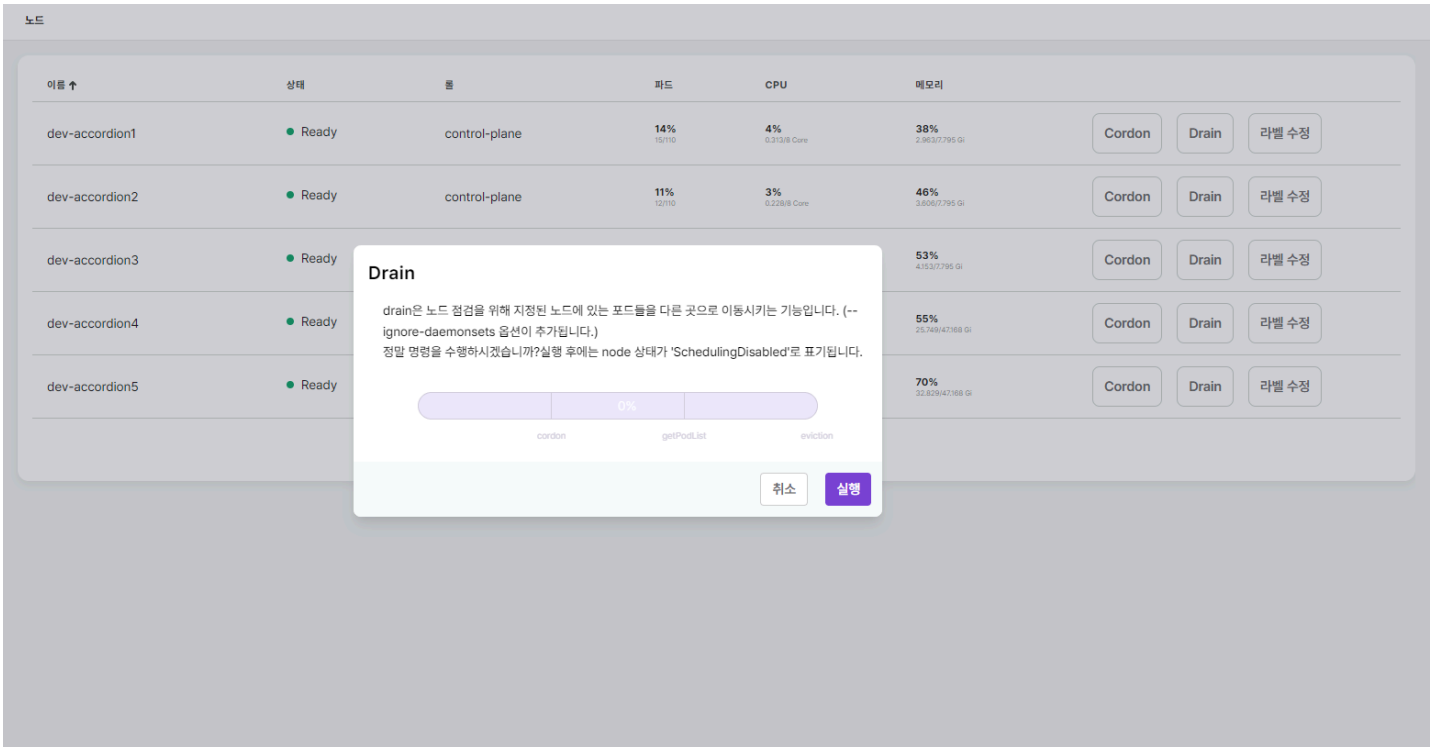

Cordon 으로 쿠버네티스 리소스 배포 대상에서 제외된 노드는 `Uncordon` 버튼을 이용해 다시 쿠버네티스 리소스 배포 대상으로 추가할 수 있다. 쿠버네티스 리소스 배포 대상으로 추가되면 `SchedulingDisabled` 롤이 제거된다.



### 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.

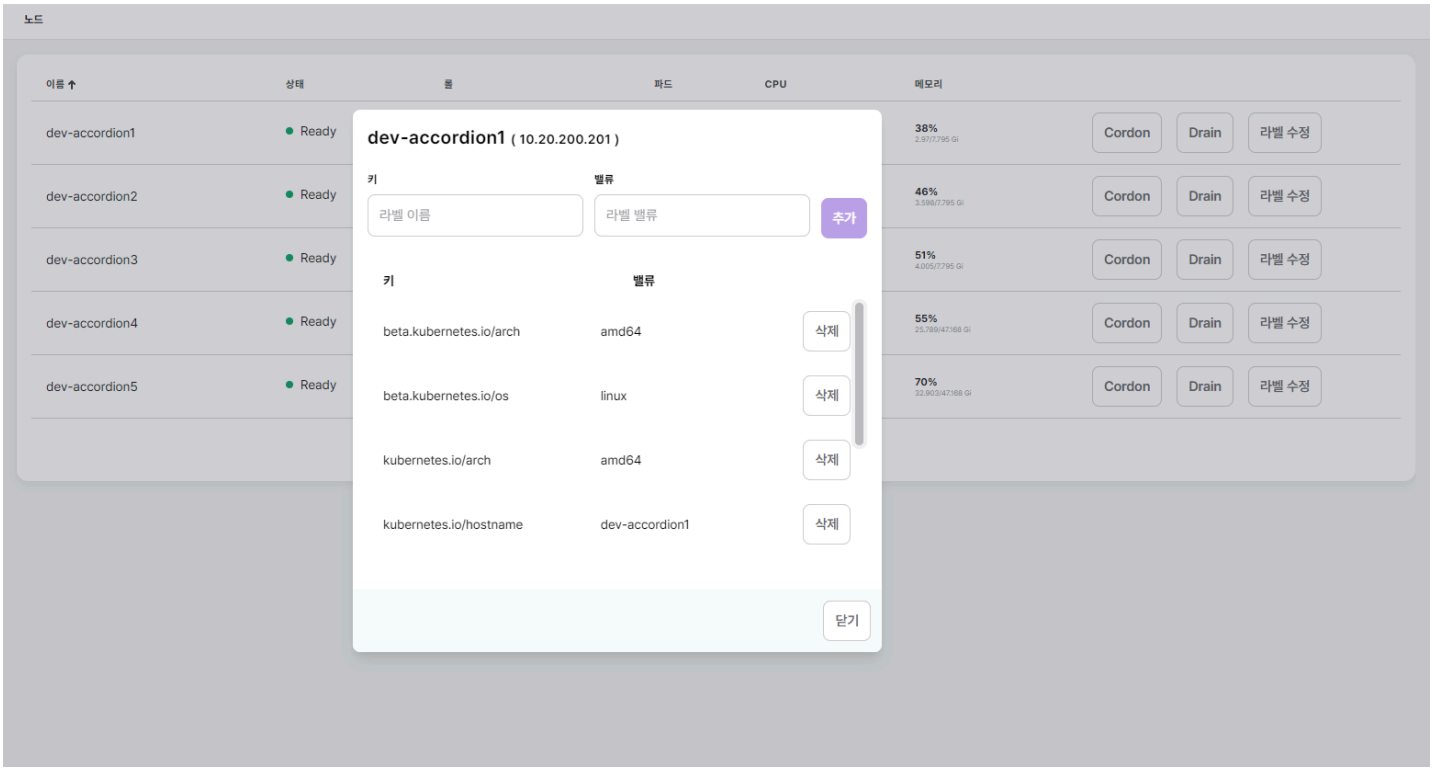
**IMPORTANT** 데몬셋으로 배포된 파드의 경우는 예외로 다른 파드로 이동하지 않는다.




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

### 4.2.3.3. 라벨 수정

노드의 레이블을 변경한다.



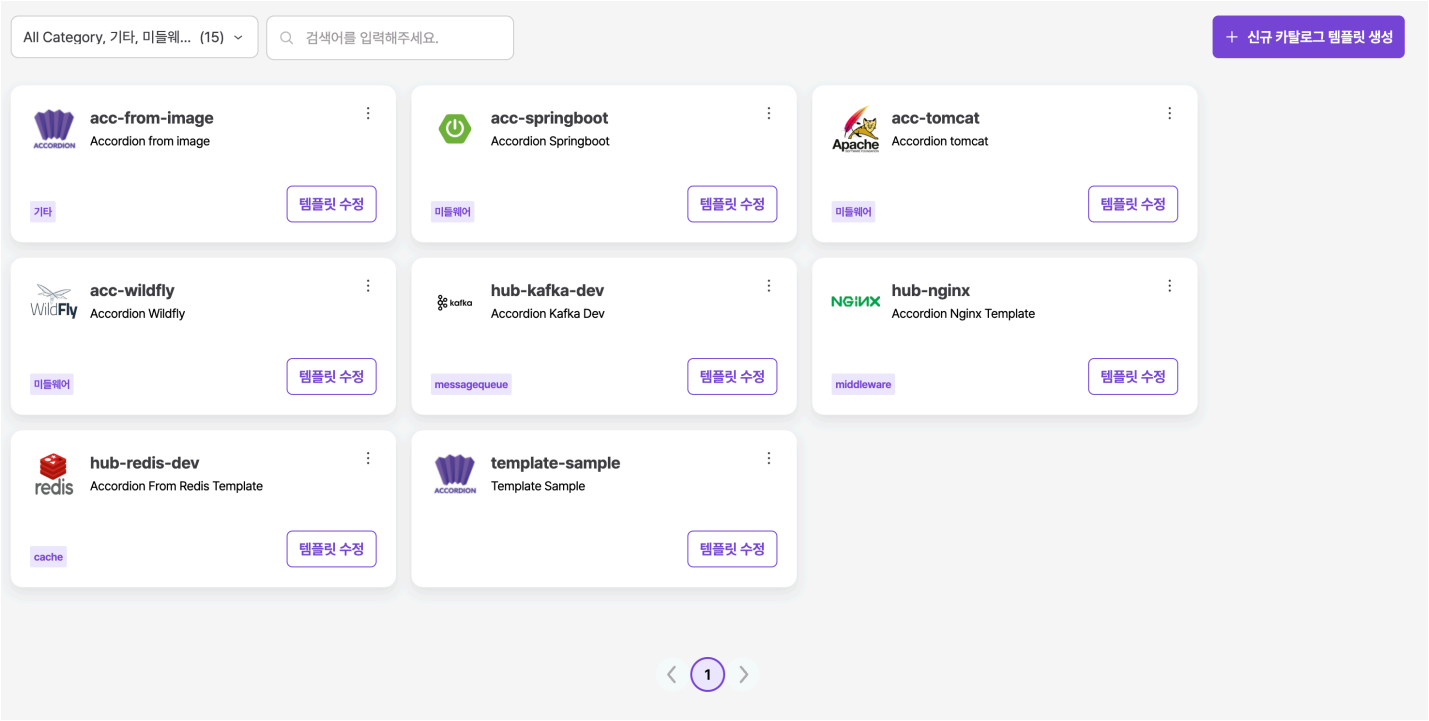
## 4.2.4. 애플리케이션

### 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**

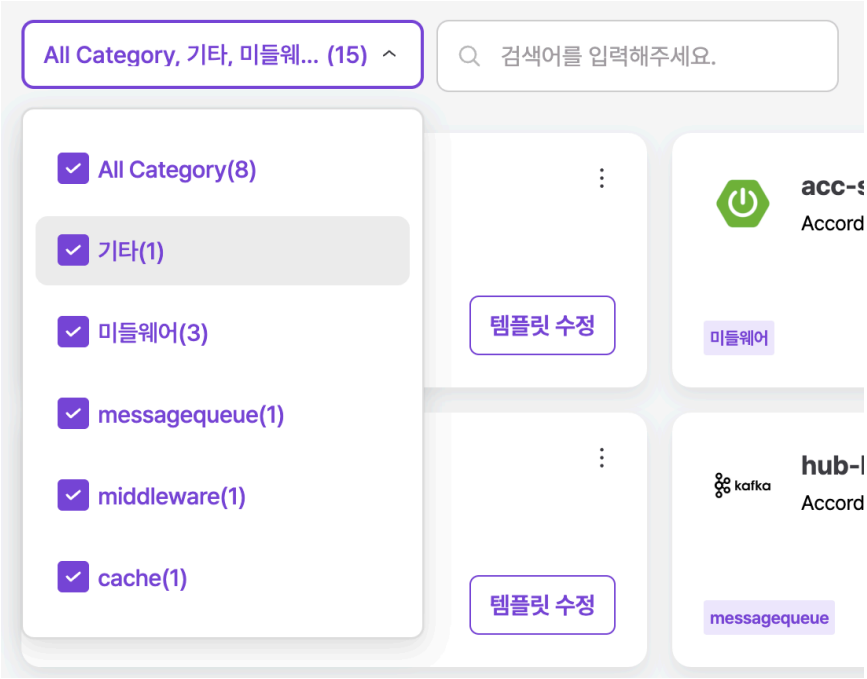
- 네임스페이스 스코프의 카탈로그 메뉴에서 카탈로그를 생성할 때 카탈로그 템플릿을 직접 생성할 수도 있다.
- 미리 카탈로그 템플릿을 만들어 두면 카탈로그를 좀 더 간단하게 생성할 수 있다.




표시되는 카드에 나타나는 정보는 다음과 같다.

| 항목 | 설명 |
|---|---|
| 이미지 | 클러스터 카탈로그 템플릿 생성 시 등록한 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 `ui.accordions.co.kr/category: <카테고리>` 형식으로 입력된 텍스트가 표시된다. |
| 스코프 | 네임스페이스 레벨인 경우 `네임스페이스` 라벨이 하단에 위치 |

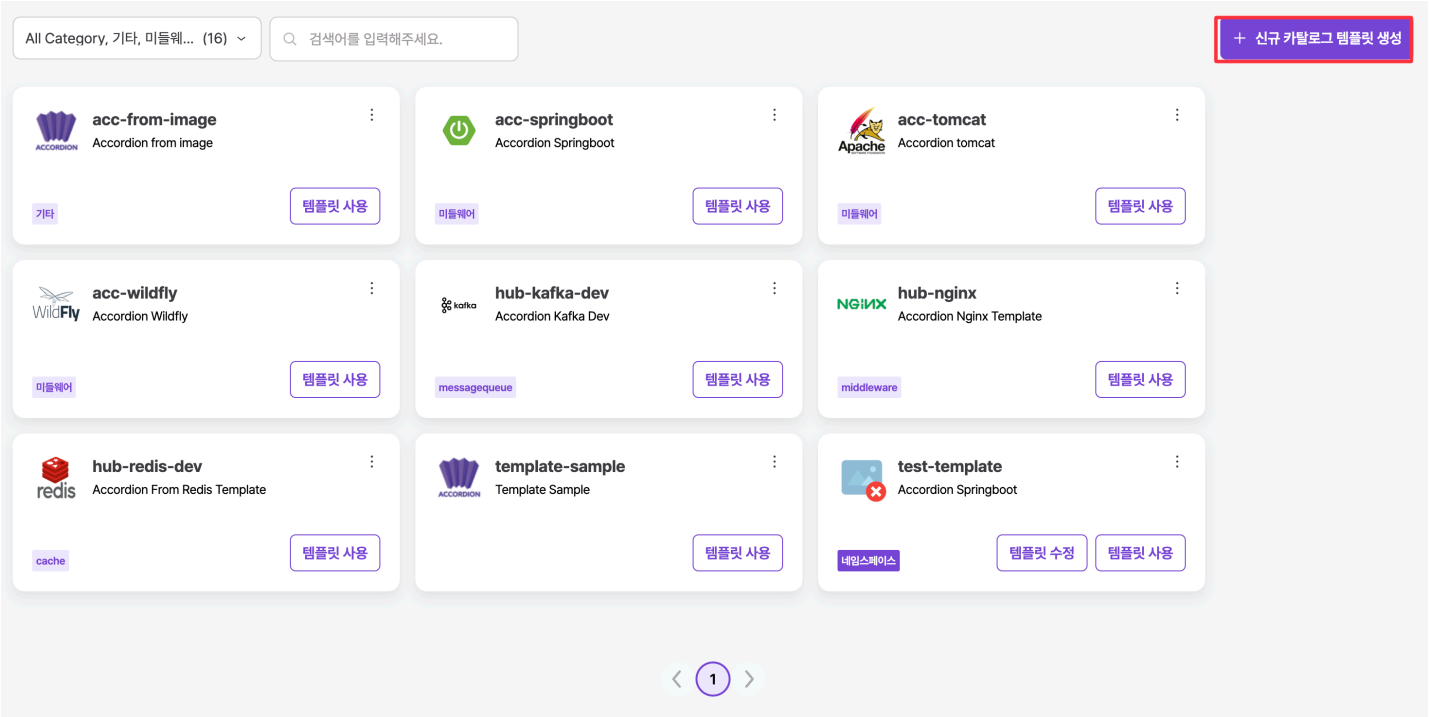
### 템플릿 카테고리 필터 및 검색




카탈로그 템플릿 카테고리 필터 및 검색 기능을 제공한다.

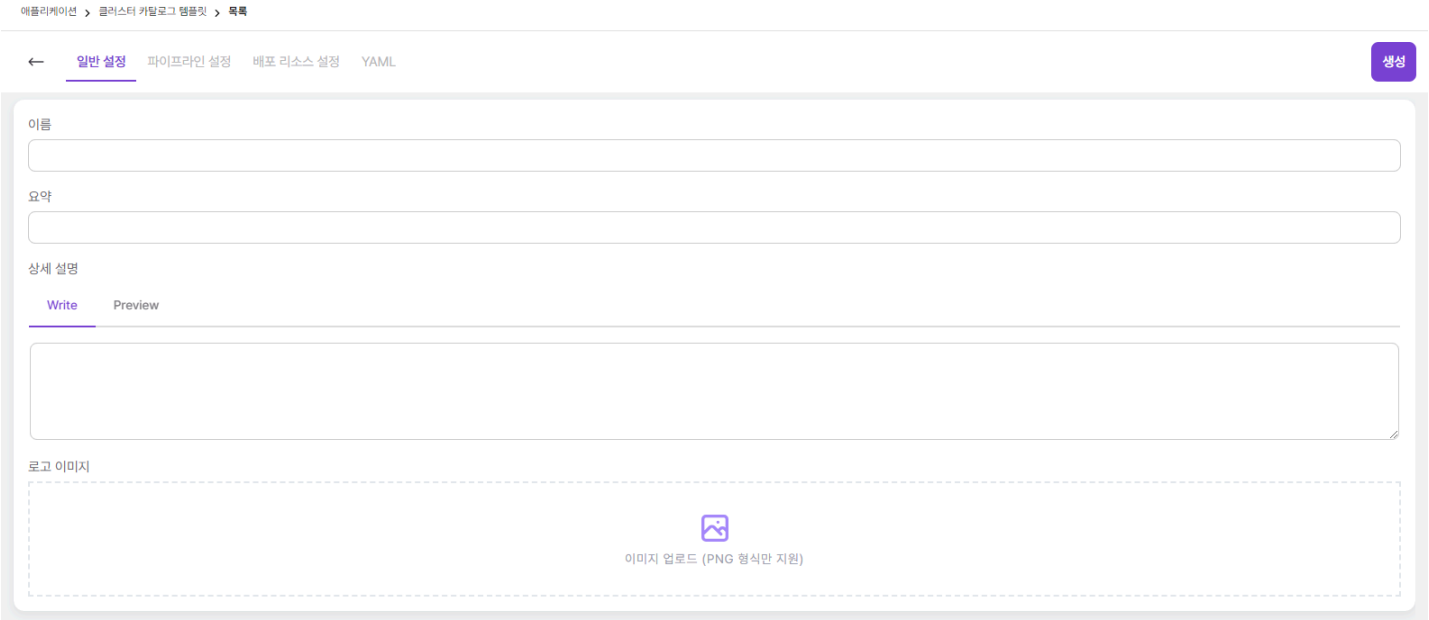
> **TIP** 카테고리는 카탈로그 YAML의 annotations 중 `ui.accordions.co.kr/category: <카테고리>` 형식으로 입력된 텍스트를 기준으로 한다.

### 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




입력하는 카탈로그 템플릿 기본 정보는 다음과 같다.

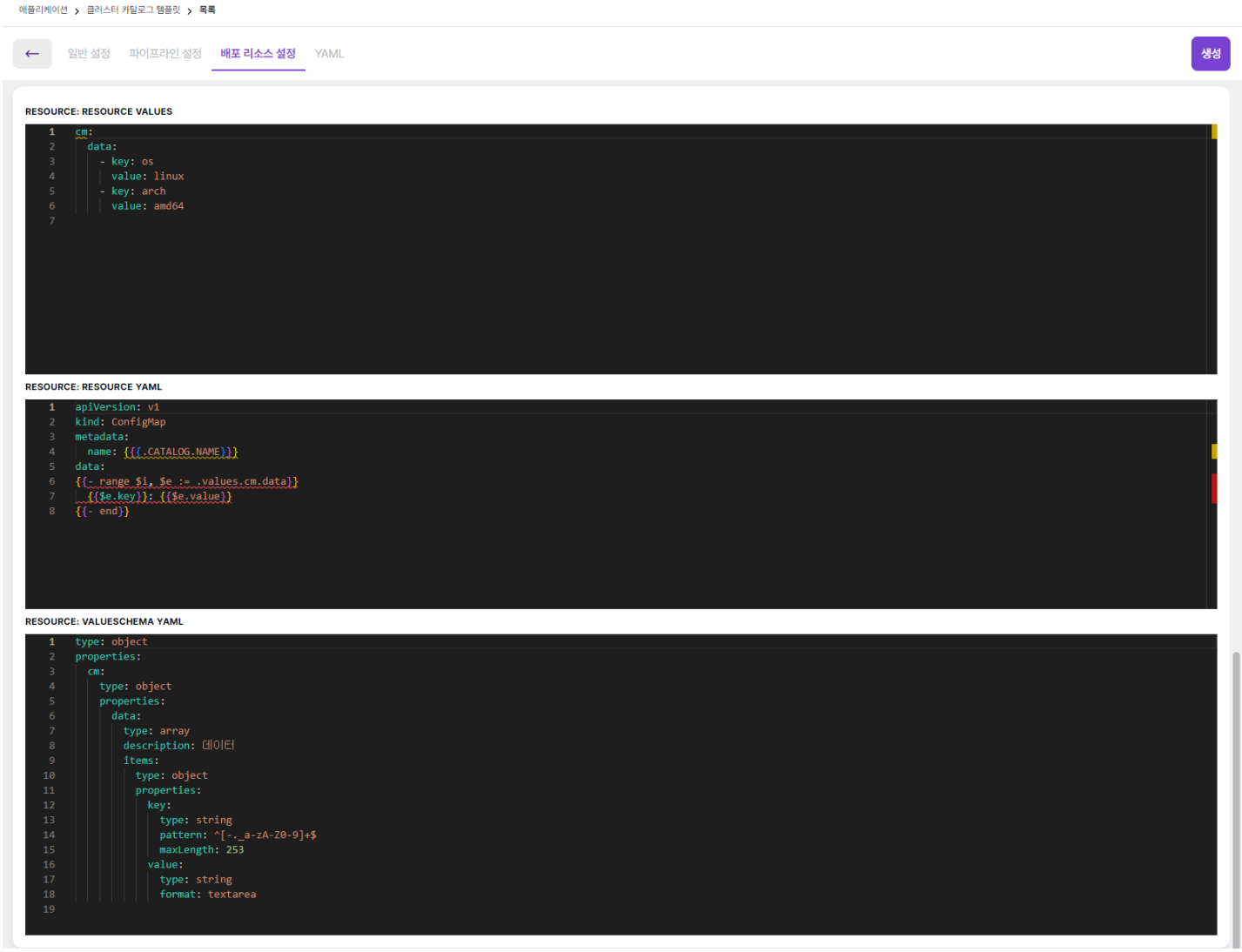
| 항목 | 설명 |
| --- | --- |
| 이름 | 클러스터 카탈로그 템플릿의 이름 |
| 요약 | 템플릿에 대한 간단한 설명(생성 후 목록 정보에 표시) |
| 상세 설명 | 템플릿에 대한 자세한 설명(생성 후 상세 설명 버튼 선택시 표시) |
| 로고 이미지 | 템플릿에 대한 로고 이미지(미설정 시 템플릿의 기본 로고로 설정) |

파이프라인 설정

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




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

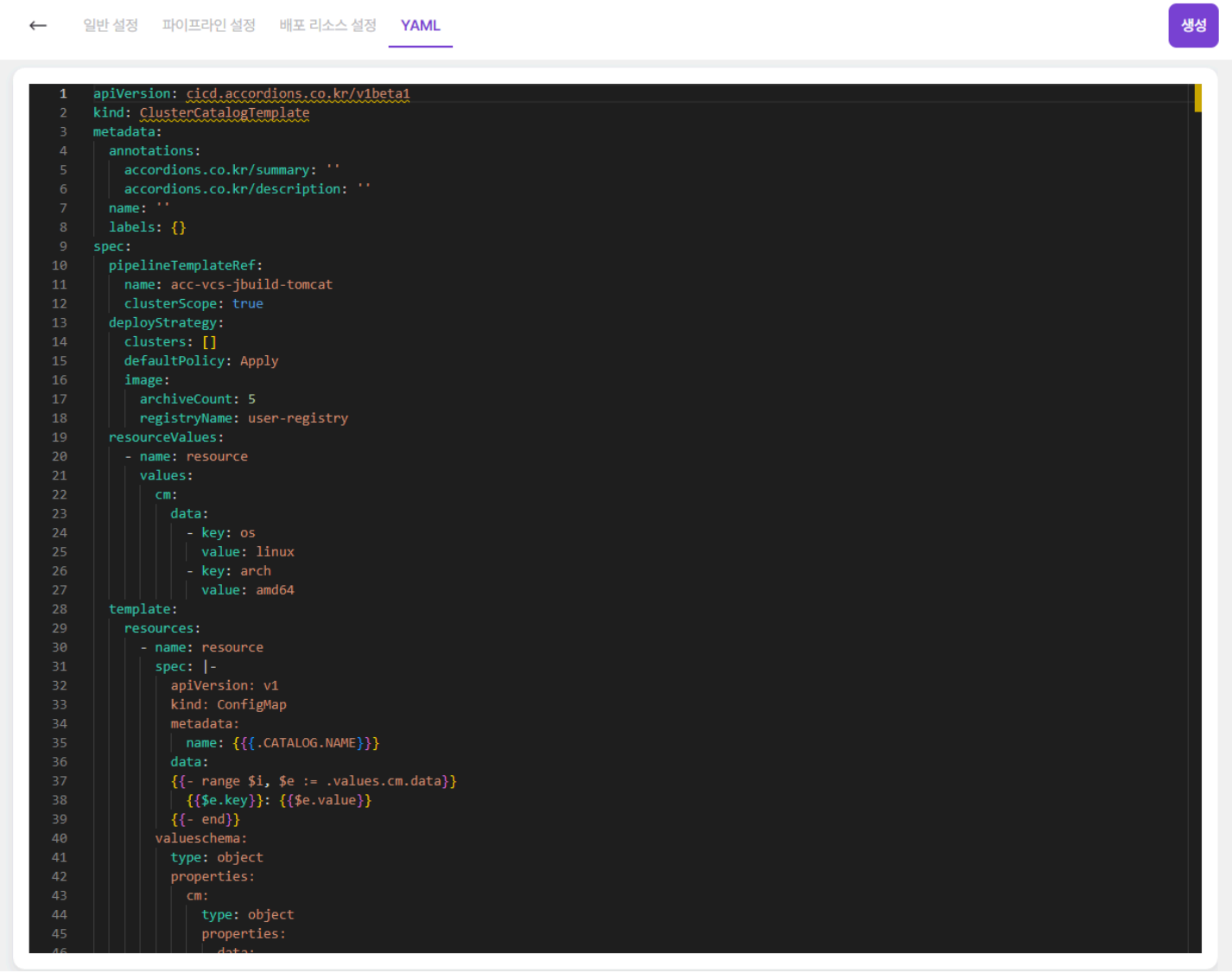
| 항목 | 설명 |
| --- | --- |
| values.yaml | 템플릿으로 전달되는 값을 정의한다. 해당 값은 resource.yaml 에서 사용되어진다. |
| resource.yaml | 쿠버네티스에 배포할 리소스들에 대한 스펙을 yaml로 정의한다. 여기에서는 카탈로그에서 사용되는 환경변수와 valueschema를 사용하여 정의할 수 있다. |
| valueschema.yaml | resource.yaml에서 사용할 valueschema를 정의한다. |

**TIP**

- valuschema의 사용법은 빌드 탭의 클러스터 태스크 템플릿의 valueschema를 참고한다.

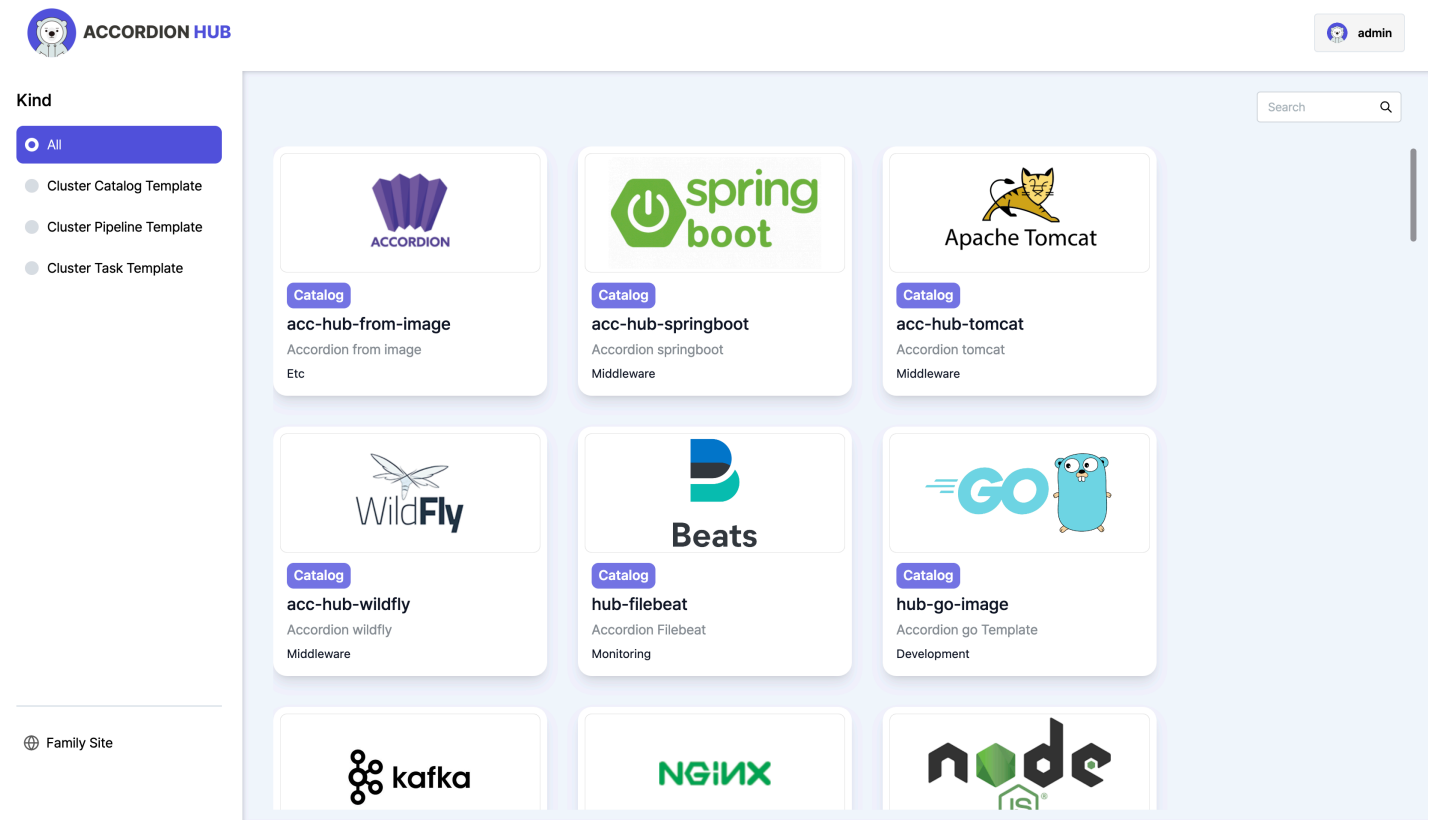
YAML

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

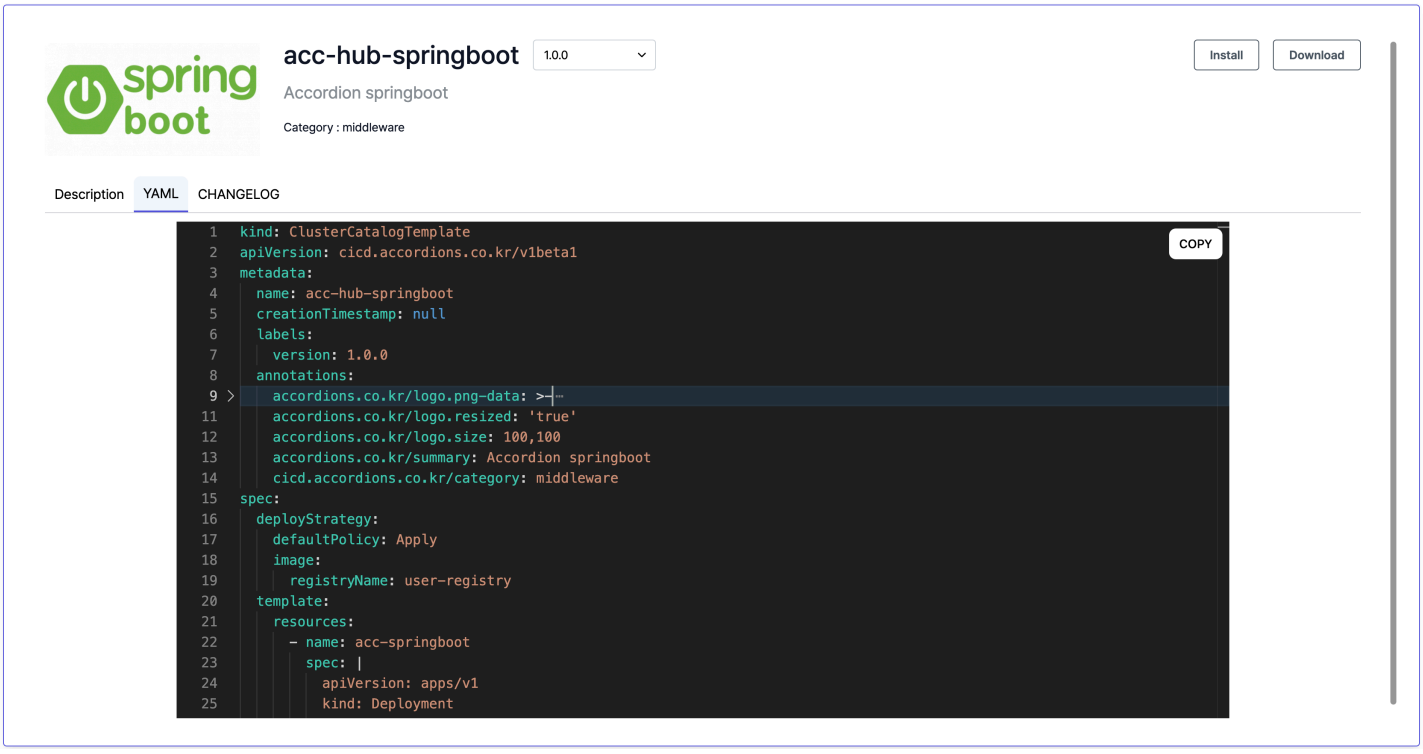


다양한 템플릿들을 제공하는 아코디언 허브 사이트 (https://hub.accordions.co.kr/) 를 운영 중으로 사이트 내의 템플릿 YAML을 복사/붙여넣기 하여 템플릿을 손쉽게 생성할 수 있다.

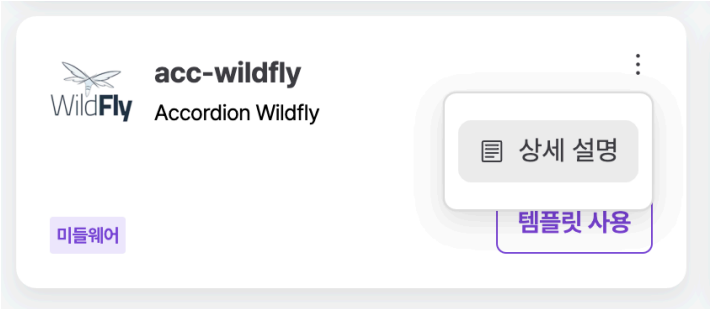
메인화면



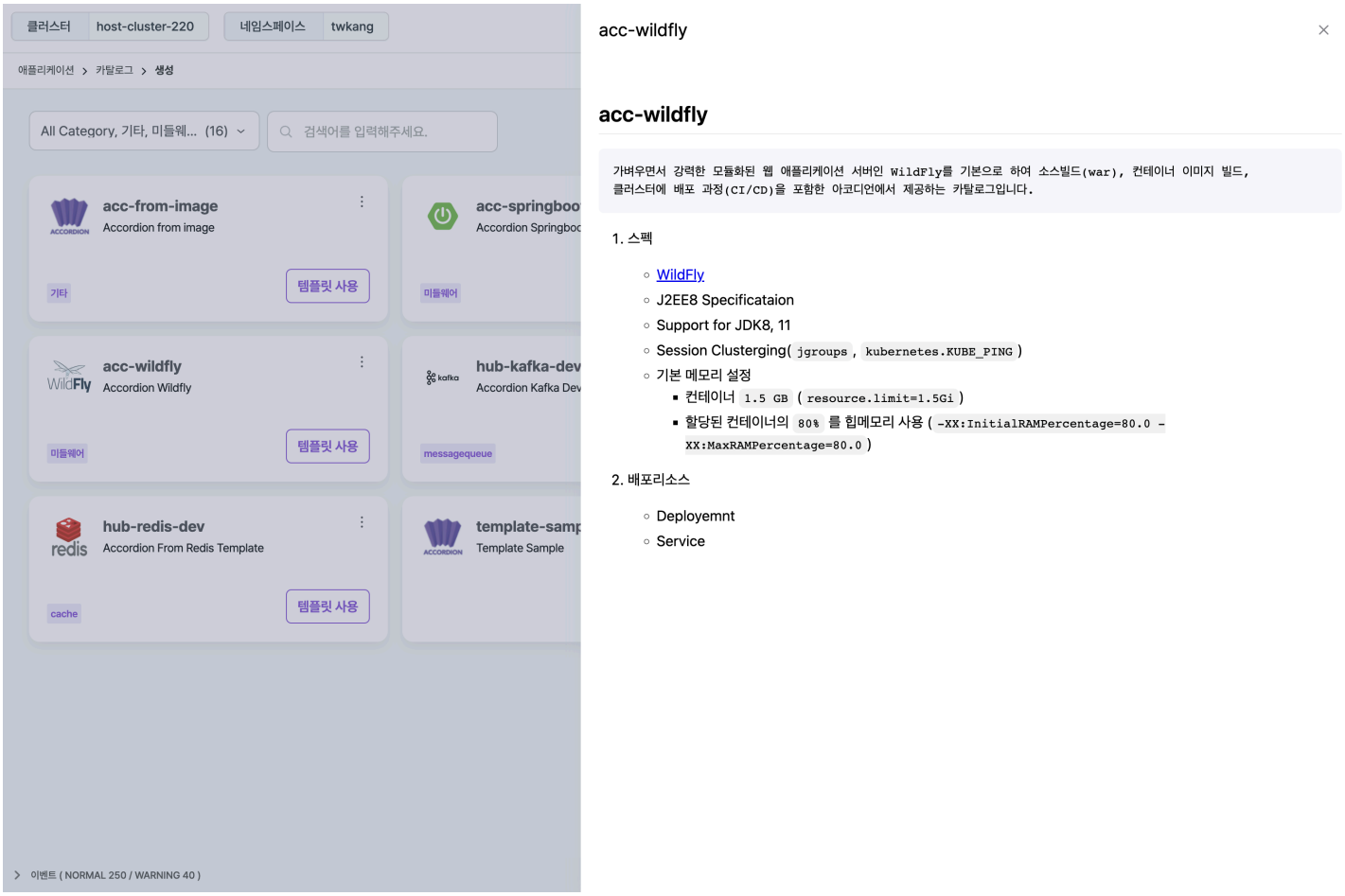

상세화면



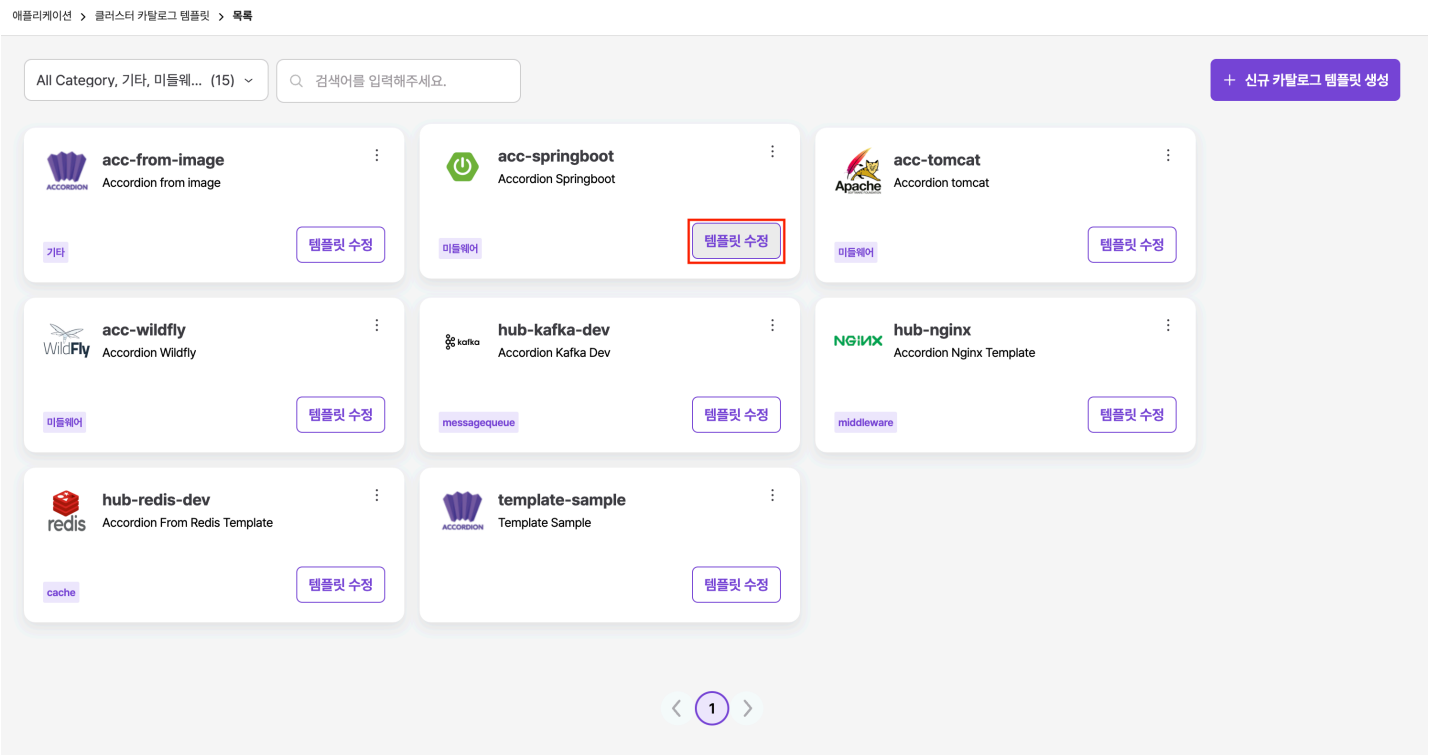
## 상세 설명



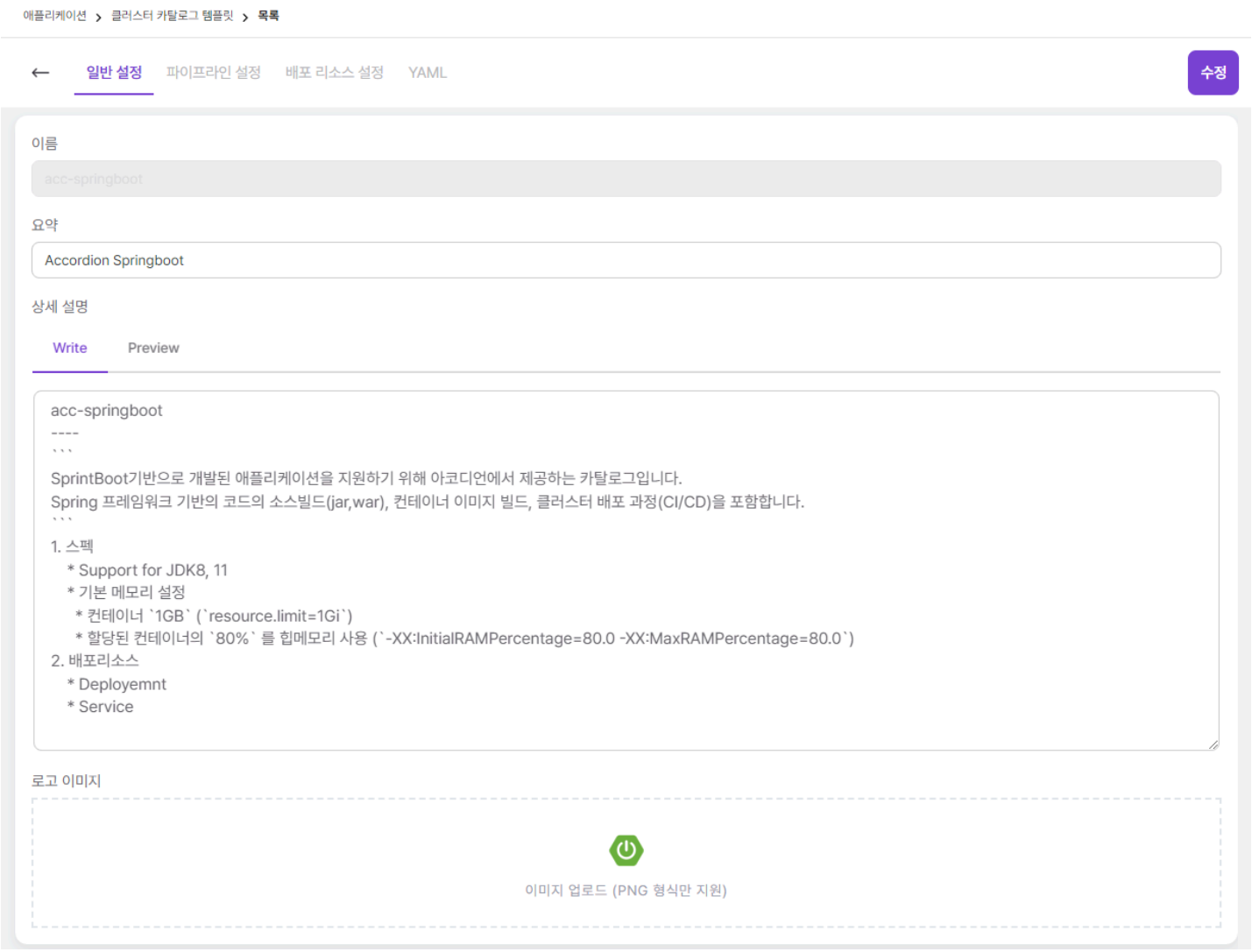

조회할 클러스터 카탈로그 템플릿 카드에 위치한 `상세 설명` 버튼을 선택하여 해당 클러스터 카탈로그 템플릿에 대한 상세 설명을 조회할 수 있다.



## 템플릿 수정

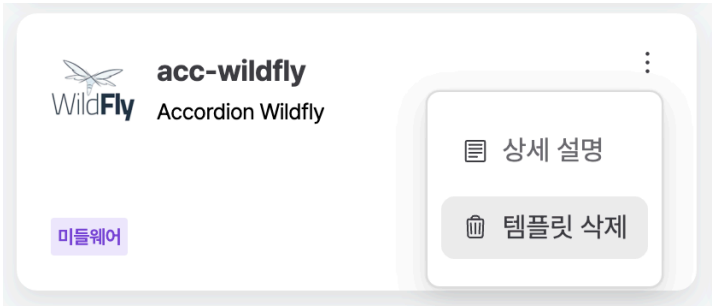



기존 클러스터 카탈로그 템플릿을 수정하기 위해서는 대상 템플릿 카드의 우측 하단에 있는 `템플릿 수정` 버튼을 선택한다.

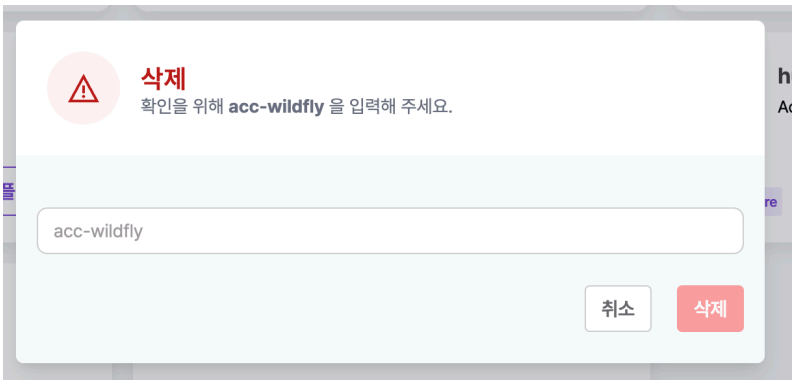


수정하길 원하는 항목을 수정한 다음 우측 상단의 수정 버튼을 선택하여 템플릿 수정을 반영한다.

## 템플릿 삭제




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



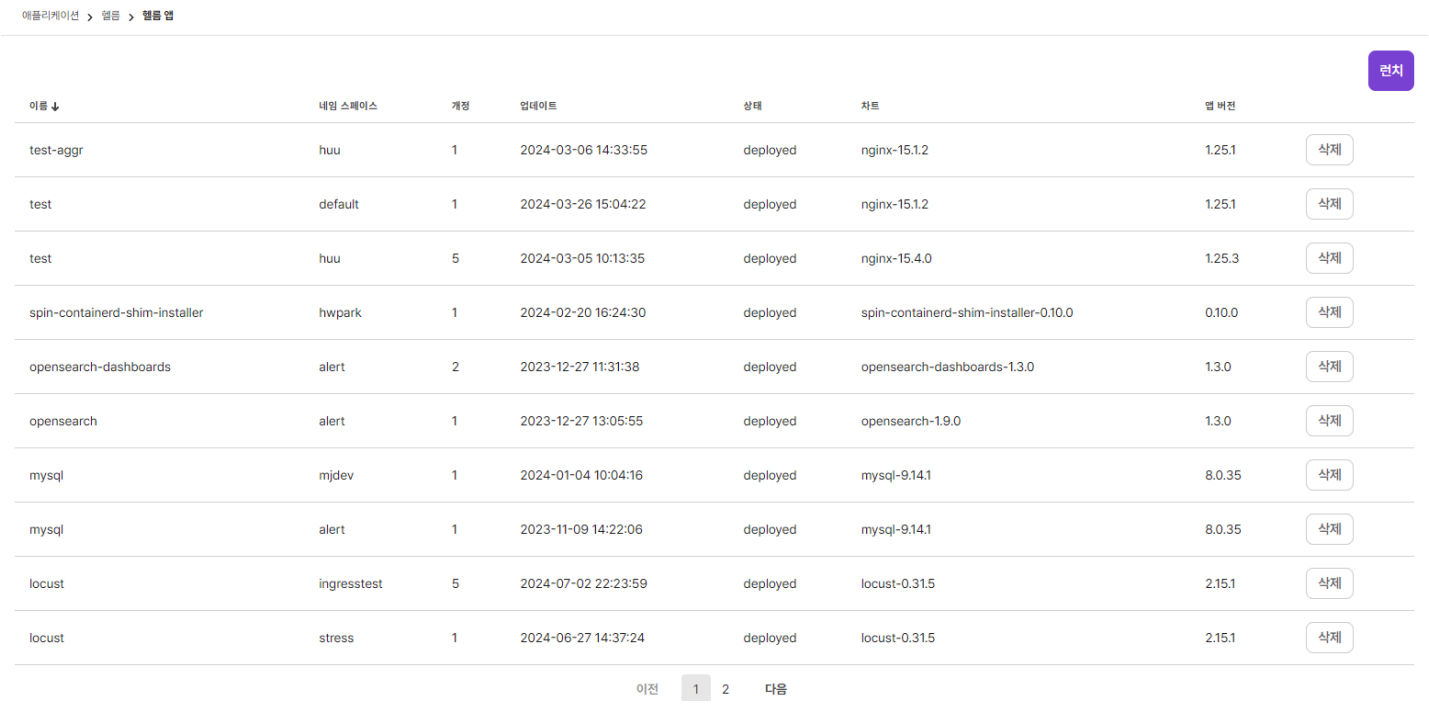

모달에서 삭제하려는 템플릿의 이름을 입력하여 삭제한다.

### 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.

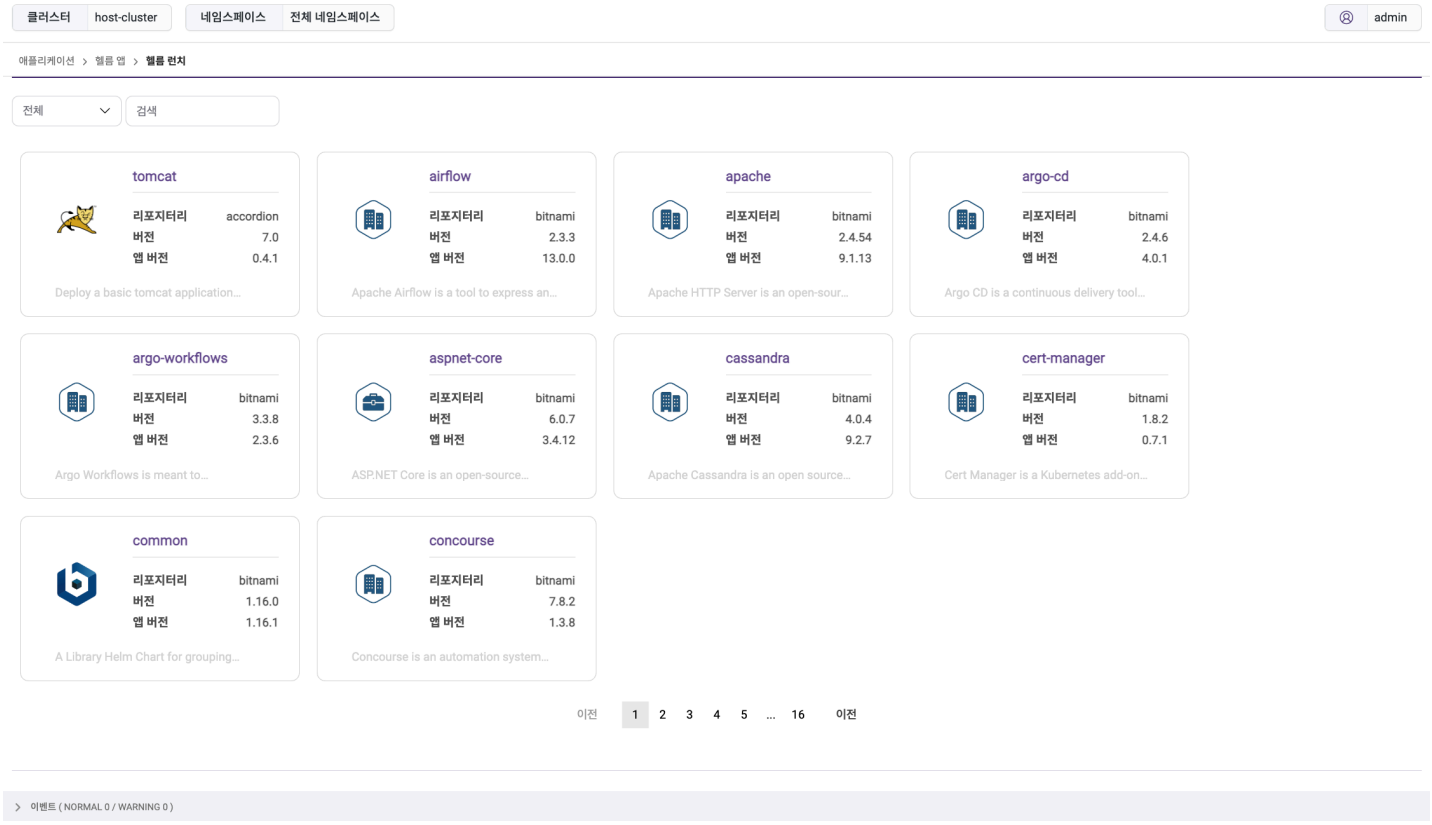



제공하는 정보는 다음과 같다.

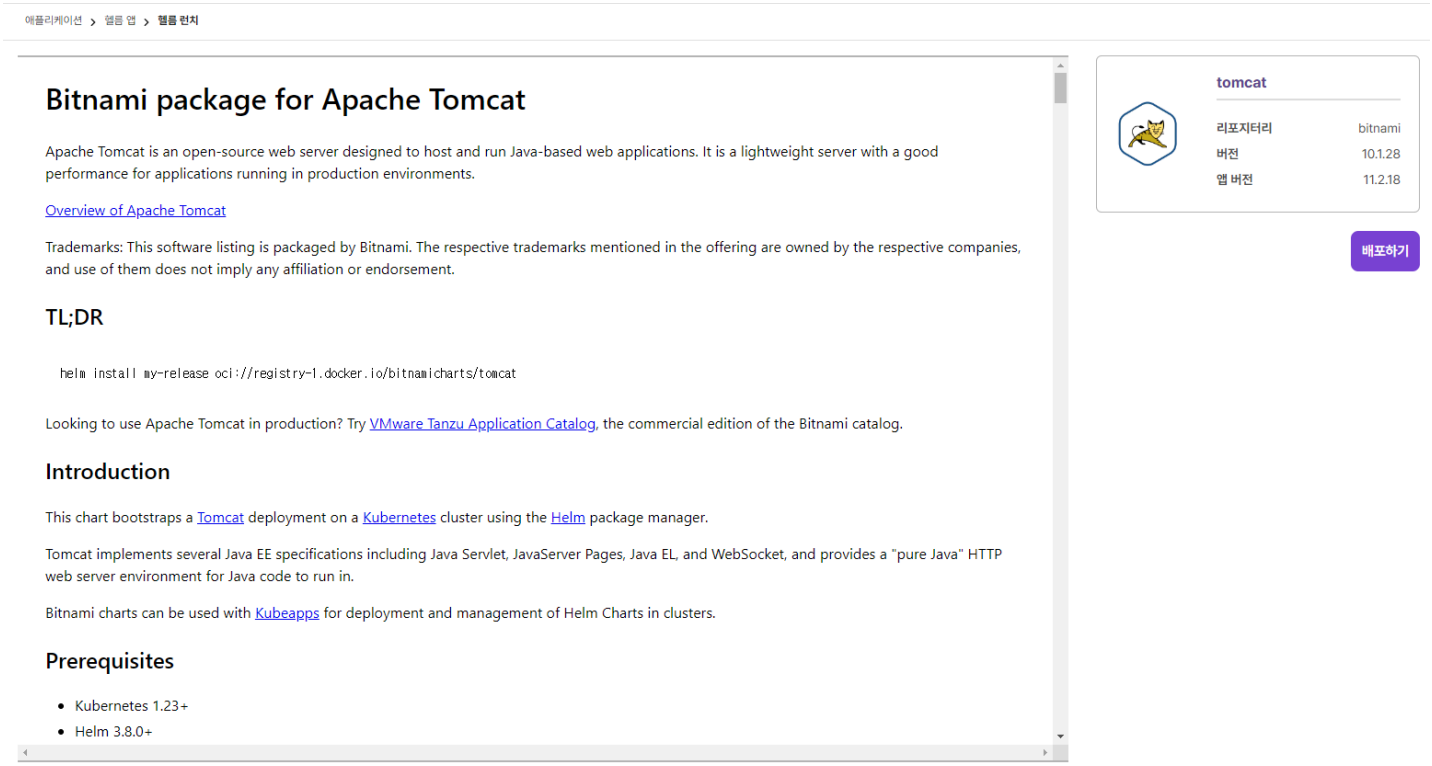
| 항목 | 설명 |
|---|---|
| 이름 | 헬름 차트를 이용해 배포한 애플리케이션 이름 |
| 네임스페이스 | 애플리케이션을 배포한 네임스페이스 |
| 개정 | 업데이트 횟수 |
| 업데이트 | 업데이트 시간 |
| 상태 | 배포 상태 |
| 차트 | 애플리케이션 배포 시 사용한 헬름 차트 |
| 앱 버전 | 헬름 차트에 설정된 앱 버전 |

## 애플리케이션 배포

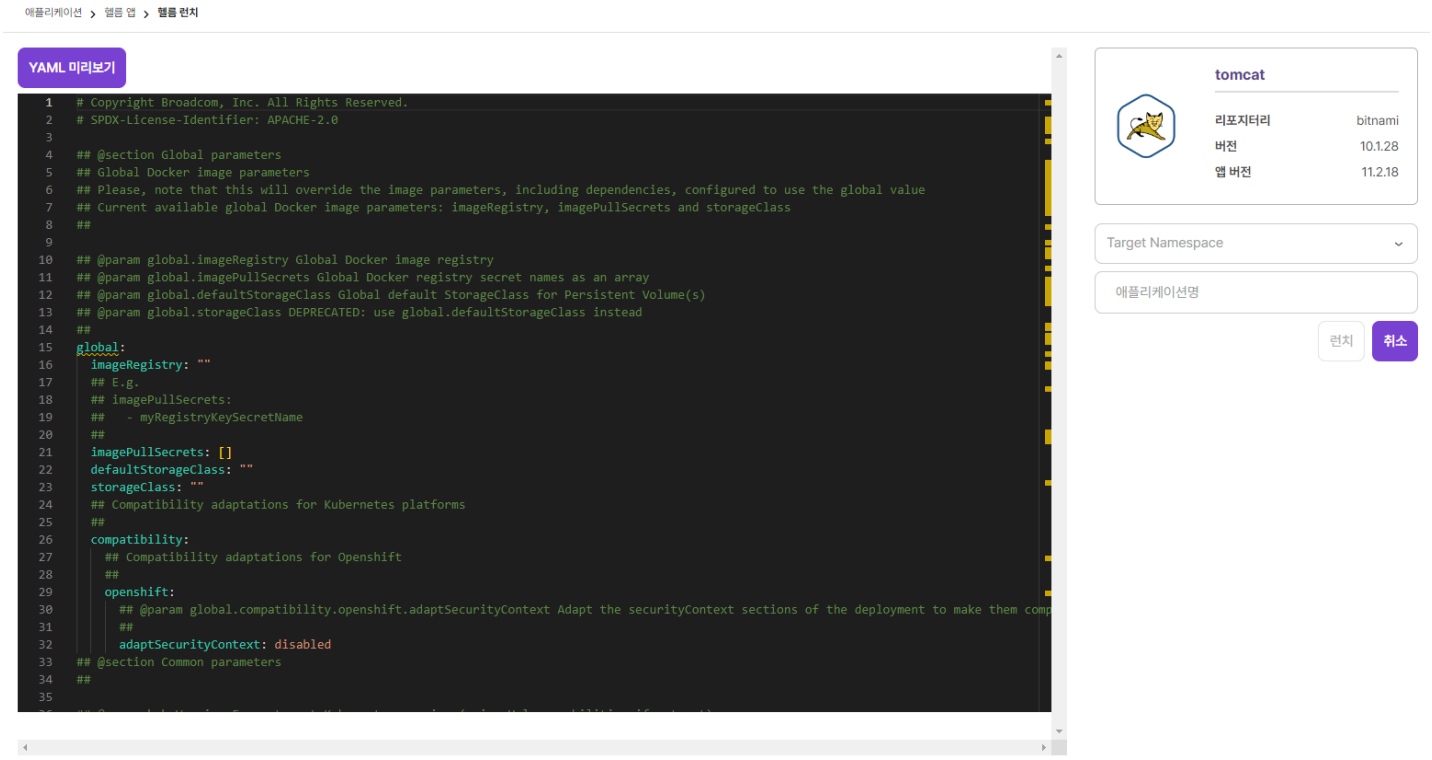
헬름으로 애플리케이션을 배포하기 위해서는 우측 상단의 런치 버튼을 선택하면 차트 목록을 조회할 수 있다.




배포할 차트를 선택하면 차트의 상세 정보를 조회할 수 있다.




배포하기 버튼을 선택하면 배포에 필요한 값을 설정할 수 있다.



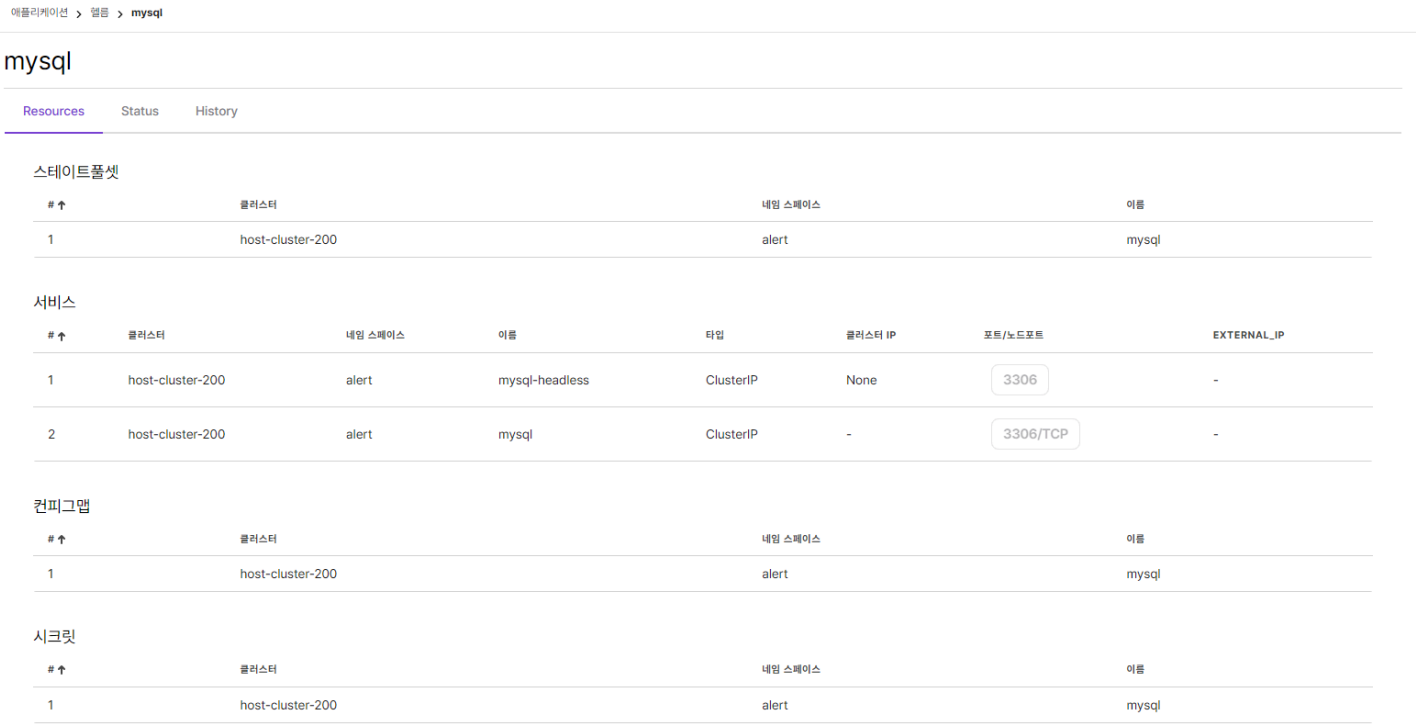

좌측의 YAML 편집기에서 차트를 애플리케이션으로 배포할 때 적용할 값을 설정할 수 있다. 우측의 `Target Namespace` 와 `애플리케이션명` 은 배포할 네임스페이스와 배포하는 애플리케이션의 이름을 설정한다. 설정이 완료되면 좌측 상단의 `YAML 미리보기` 버튼을 눌러 배포될 쿠버네티스 리소스를 미리 확인할 수 있다.

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

작성을 완료하여 `런치` 버튼을 선택하면 실제 클러스터에 애플리케이션이 배포된다.

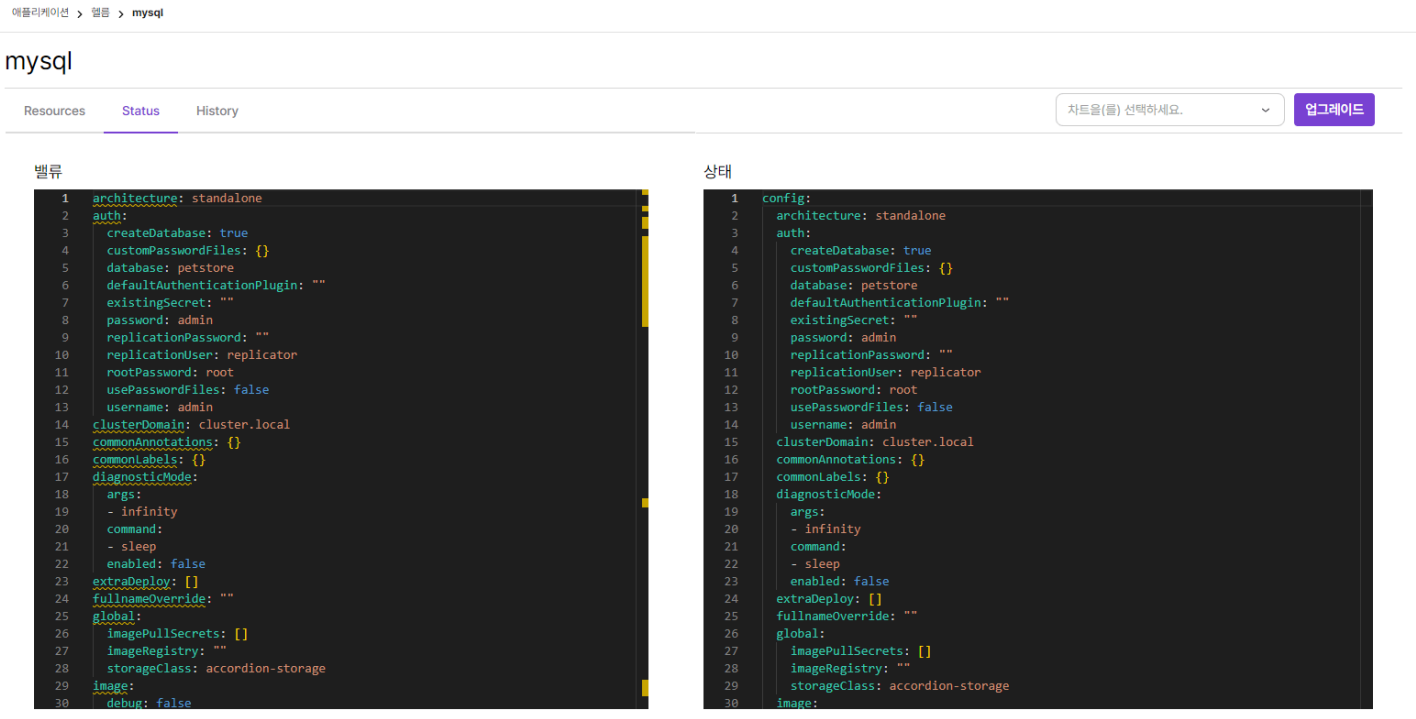
## 애플리케이션 정보 조회

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

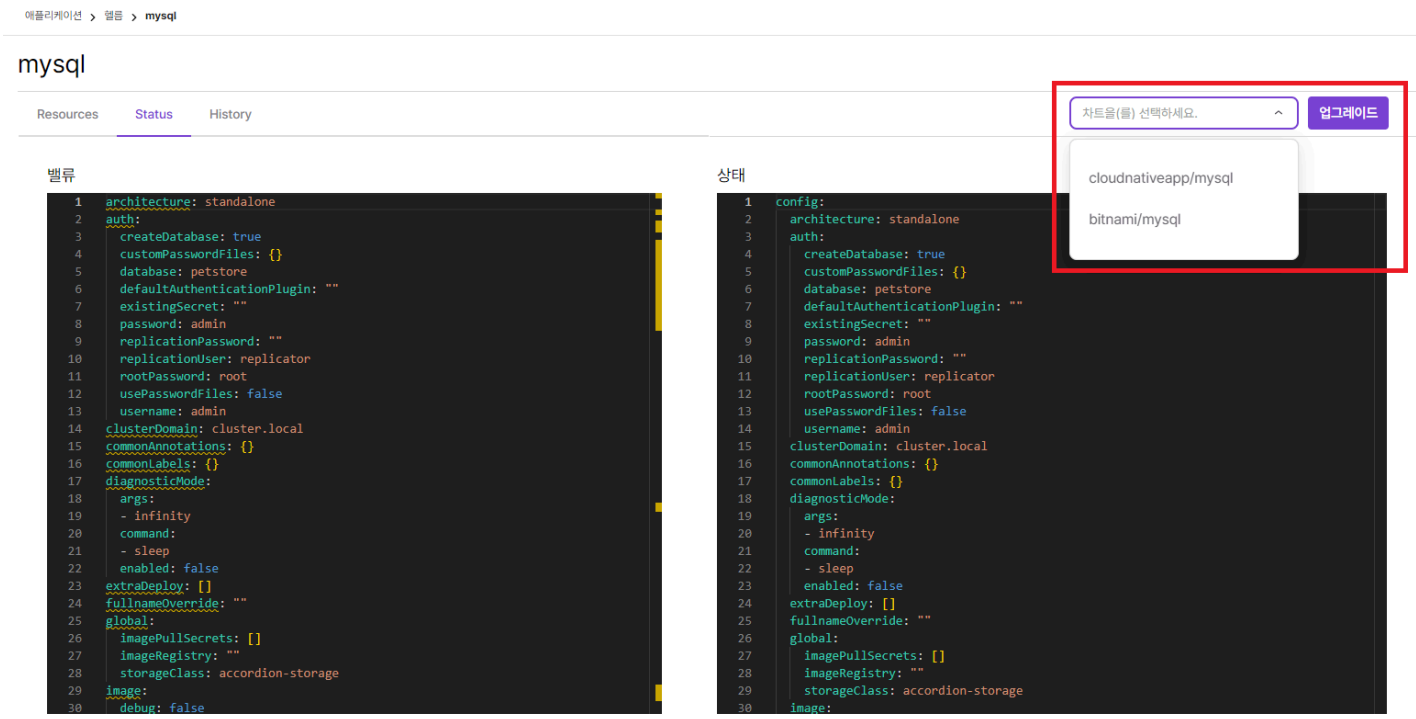



리소스 탭에서는 애플리케이션을 구성하는 쿠버네티스 리소스들에 대한 정보를 제공한다.

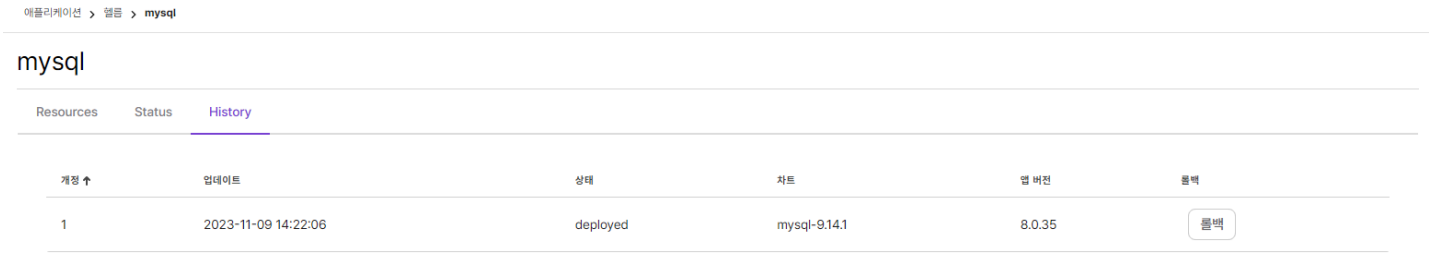
> **TIP** 리소스 정보는 애플리케이션 구동에 필수적인 파드와 서비스 등의 정보에 대해서만 조회가 가능하다.




상태 탭에서는 설정값과 배포 상태 정보를 제공한다.



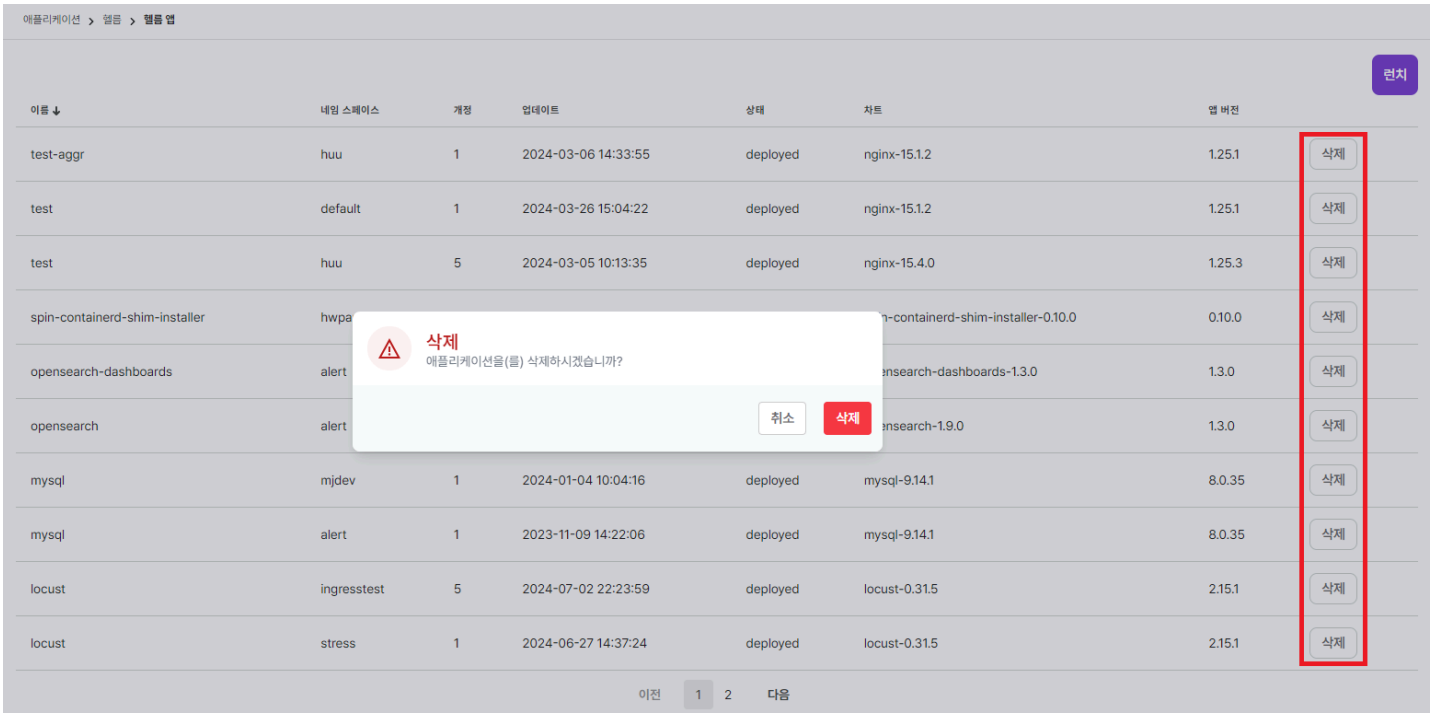

우측 상단의 차트 선택 후 업그레이드 버튼을 클릭해 배포된 애플리케이션을 업그레이드할 수 있다.




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.



## 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 | configmap존재 여부만 확인하며, metadata.name을 참조 |
| 쿠버네티스 | configmapData | configmap의 data의 요소를 참조 |
| 쿠버네티스 | secretName | secret존재 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.

YAML

```yaml
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
```

**bDefaultVal**
기본값이 있는 boolean타입

true

**bEnumVal**
boolean 타입, enum : 값의 범위 지정

true

**bVal**
기본형: boolean타입

false

number
입력값을 숫자 형식으로 제한한다.

YAML

```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: number타입"
        maximum: 100 # (옵션) 해당 값보다 크다면 오류
        minimum: 1   # (옵션) 해당 값보다 작다면 오류
      nDefaultVal:
        type: number
        description: "기본값이 있는 number타입"
        default: -1
```

**nDefaultVal**
기본값이 있는 number타입

```
-1
```

**nVal**
기본형: number타입

```
0
```

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 |

YAML

```yaml
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)

mm/dd/yyyy

**email**
email 형식

**ip**
ip 형식

**sshPublic**
ssh public 형식

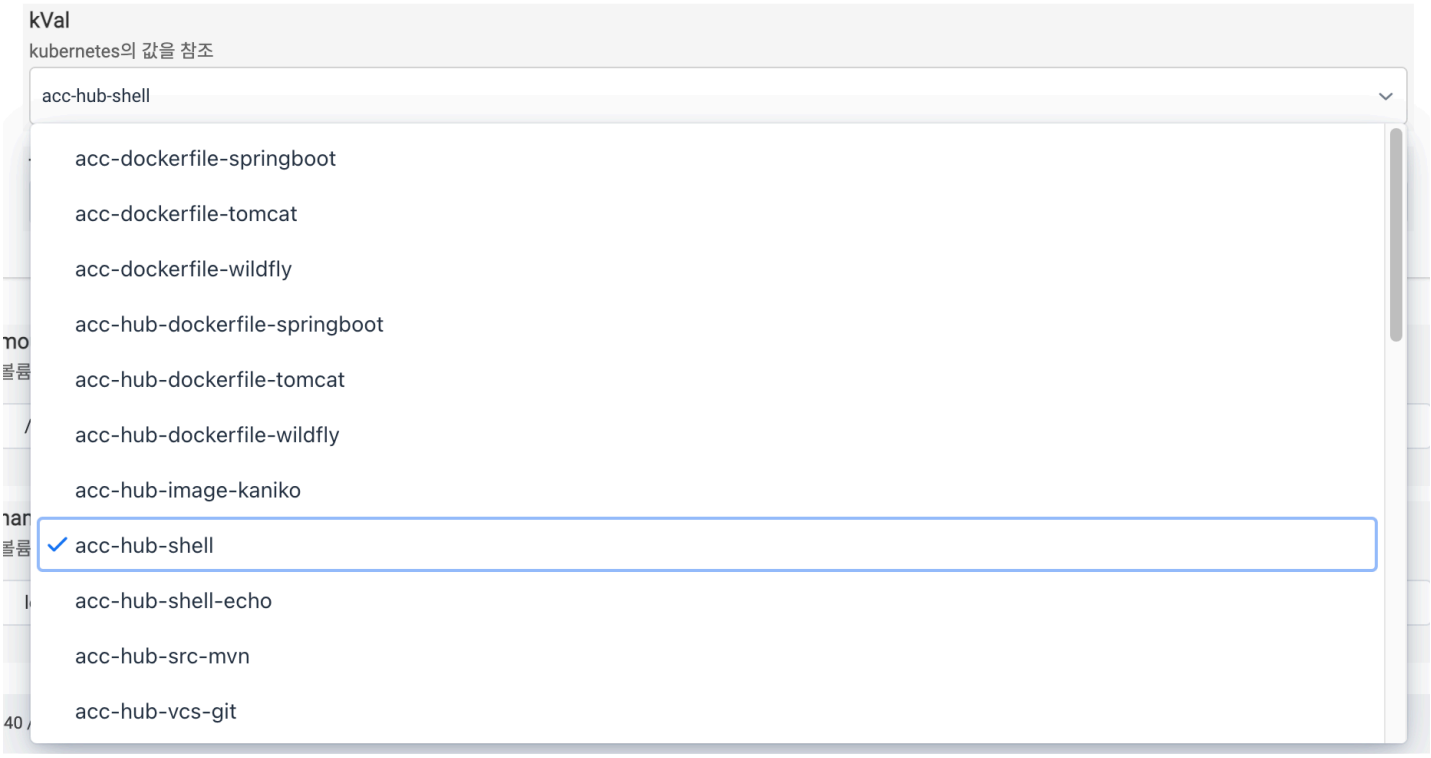
**textarea**
멀티라인 형식 지원

## valueschema의 쿠버네티스 유형

kubernetes
입력값을 쿠버네티스 리소스를 선택하도록 한다.

YAML

```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("")
```

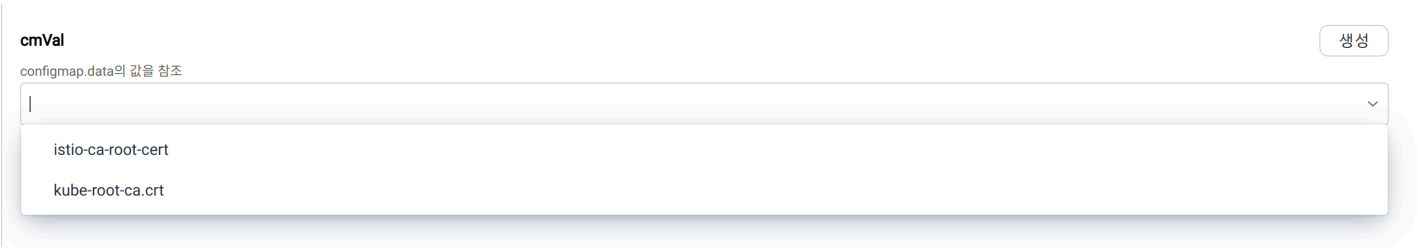


### configmapData

입력값을 configmap 리소스를 선택하도록 하며, data 필드를 참조한다.

YAML

```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("")
```

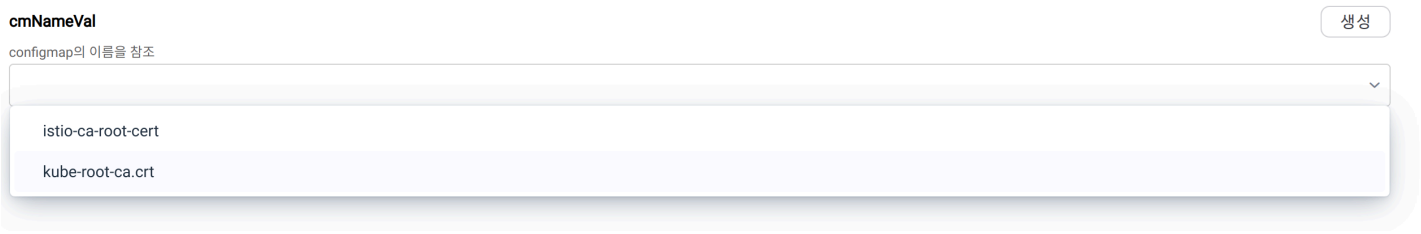


### configmapName

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

YAML

```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 셀렉터
        x-kube-optional: true                 # (옵션) true이면서 사용자 입력값이 없다면 공백("")
```

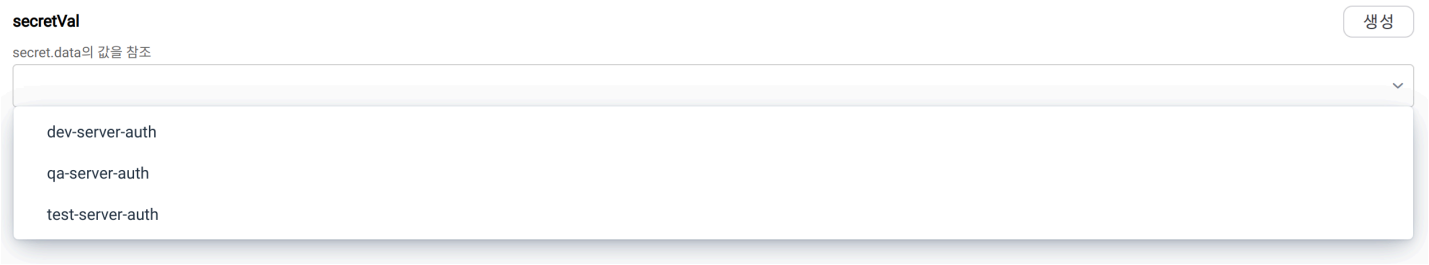


### secretData

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

YAML

```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) scecret.type과 일치하는지 유효성 검사
        x-kube-labelSelector: ""               # (옵션) 라벨 셀렉터
        x-kube-optional: true                  # (옵션) true이면서 사용자 입력값이 없다면 공백("")
```

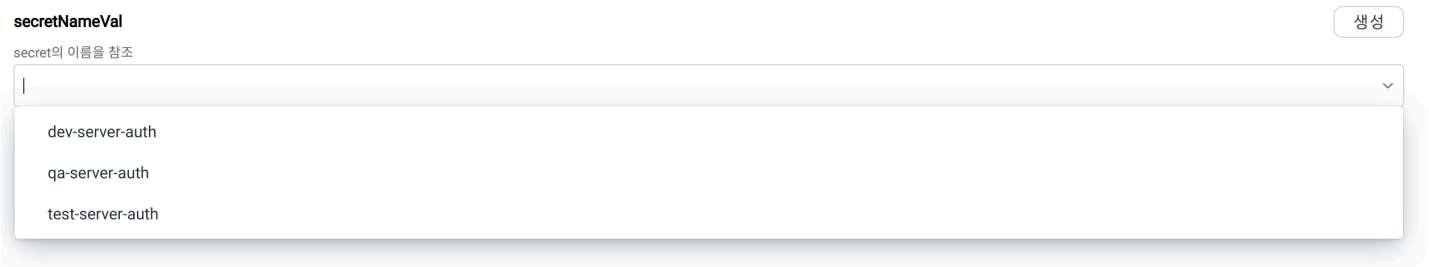


### secretName

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

YAML

```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) scecret.type과 일치하는지 유효성 검사
        x-kube-labelSelector: ""                # (옵션) 라벨 셀렉터
        x-kube-optional: true                   # (옵션) true이면서 사용자 입력값이 없다면 공백("")
```



## valueschema의 참조 유형

array
입력값을 배열 형식으로 받는다.

YAML

```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 ₩"₩telement[" $i "]: " $e "₩";"}}
        {{- 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

```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

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

        {{- else if eq .values.secret.type "kubernetes.io/ssh-auth"}}
        {{print "echo ₩"" .values.secret.type "₩";"}}
        {{print "echo ₩"" .values.secret.privatekey "₩";"}}
        {{- 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)

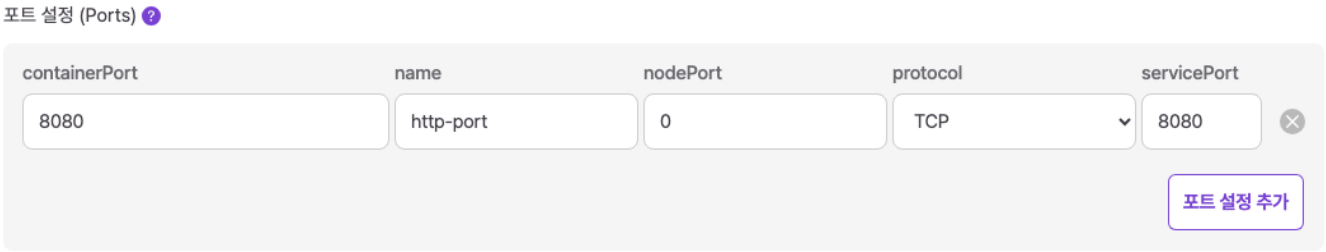
- list
  - 기본 목록 방식으로 각 properties를 행마다 표시한다.

- 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

YAML

```yaml
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
```




위처럼 각 property 가 한 행으로 지정한 비율에 맞추어 표시된다.

사용 예시 2: Labels 설정

YAML

```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
```

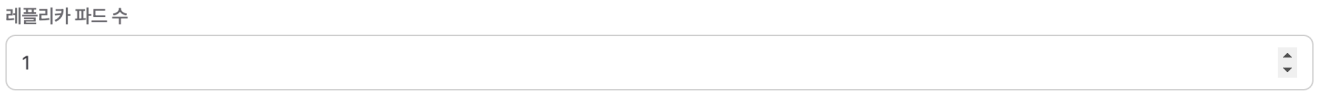
자주 사용되는 key-value 형태의 UI 형식으로 labels, annotations 등에 사용된다.

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

사용 예시: Replicas 설정

YAML

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

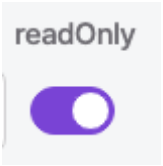



- switch
  - `type: boolean` 인 경우 사용 가능한 옵션으로, 해당 필드의 boolean 값을 변경할 수 있는 UI를 제공한다.

사용 예시: ReadOnly 설정

YAML

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




- 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

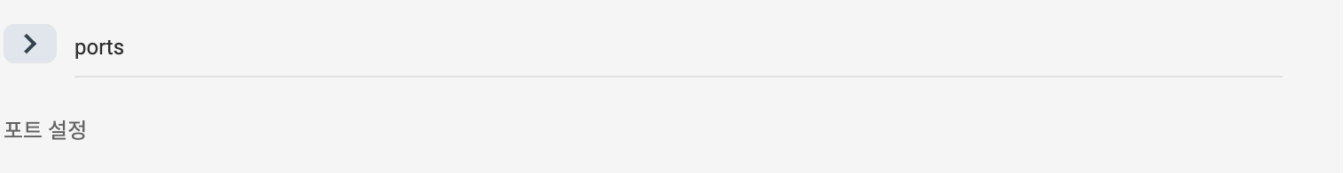
```yaml
imagePullPolicy:
  default: Always
  description: 이미지를 가져오는 전략
  enum:
    - Always
    - IfNotPresent
    - Never
  type: string
  x-ui-format: radio
```

x-ui-collapsable
Collapse, Expand 버튼이 표시되어, 해당 기본값을 지정할 수 있는 옵션을 정의한다.

- true: 숨김
- false: 표시

사용 예시

```yaml
ports:
    ...
    x-ui-collapsable: true
```




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
```

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

사용예시

RESOURCE YAML

YAML

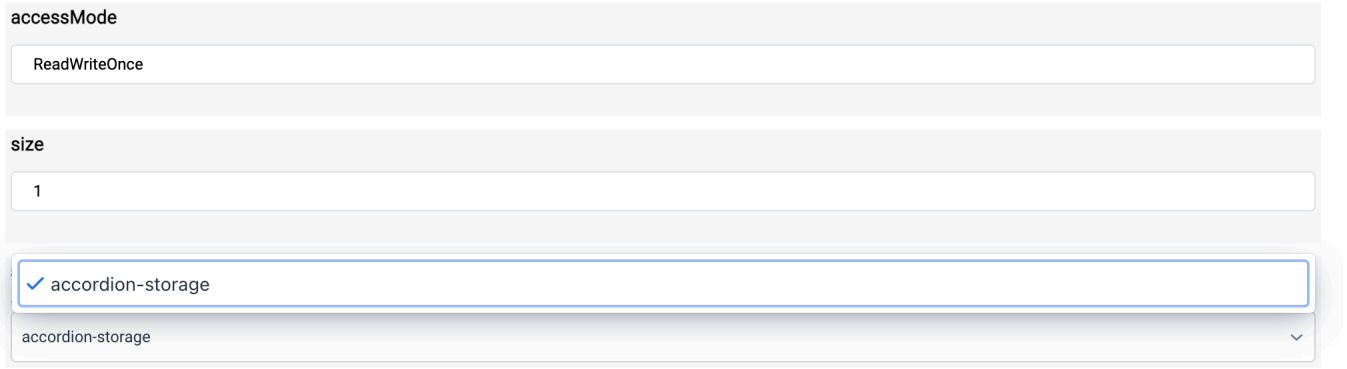
```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: {{{.CATALOG.NAME}}}
spec:
  accessModes:
  - {{.values.accessMode}}
  resources:
    requests:
      storage: {{.values.size}}
  storageClassName: {{.values.storageClass}}
  volumeMode: Filesystem
```

VALUESCHEMA YAML

YAML

```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




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.

YAML

```yaml
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
```

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

태스크의 이름은 쿠버네티스 이름 정책에 맞춰 작성하고 태스크의 연관관계는 해당 태스크을 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.

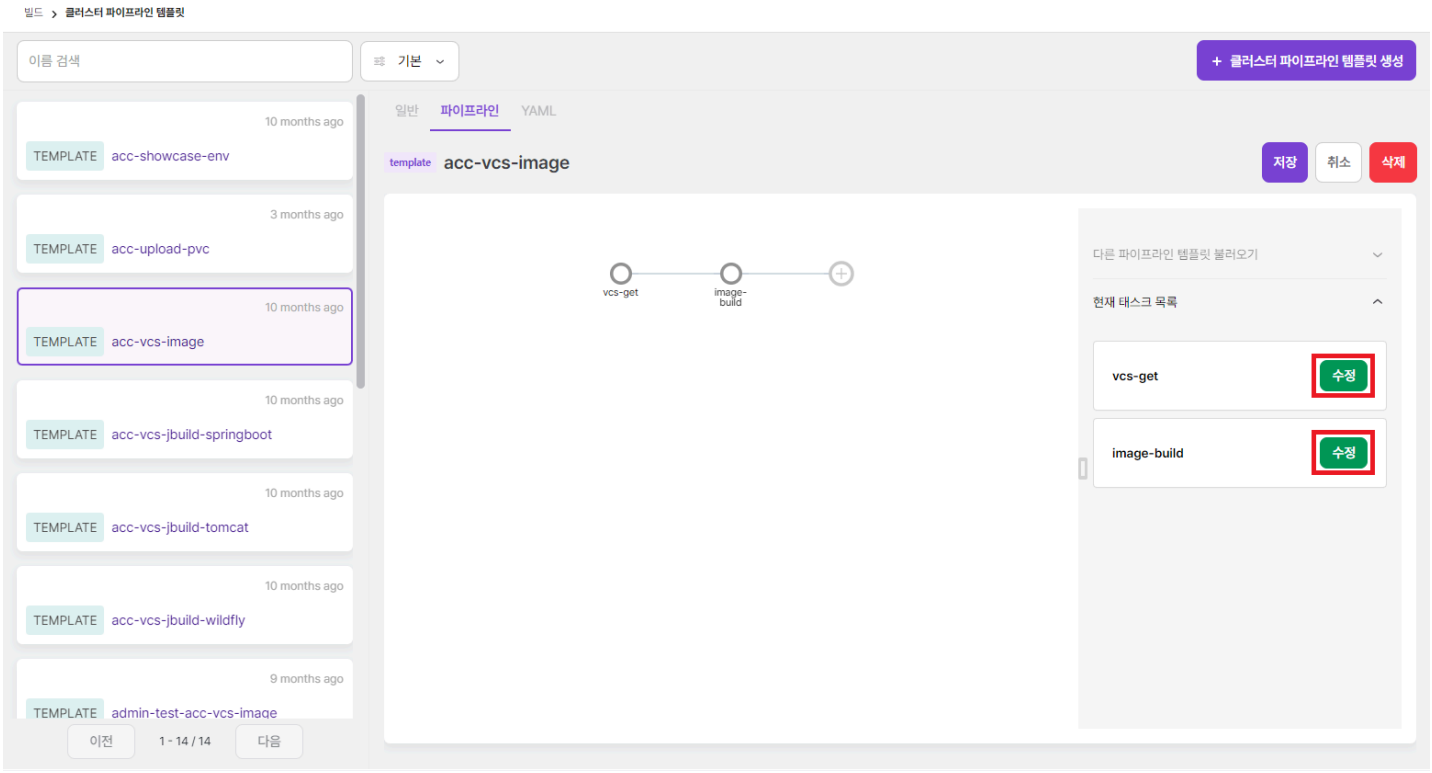
- 태스크의 `{이름}.{상태}` 형식으로 입력하며 상태는 필수로 입력해야 한다.
- `&&`, `||` 를 추가하여 논리적인 표현도 작성 가능하다.
  - `&&` 는 and 를 뜻하는 표현으로 `A.{상태} && B.{상태}` 인 경우 A 상태와 B 상태 모두인 경우를 표현한다.
  - `||` 는 or 를 뜻하는 표현으로 `A.{상태} || B.{상태}` 인 경우 A 상태 또는 B 상태 중 하나를 표현 한다.
- `({이름}.{상태} || {이름}.{상태}) && {이름}.{상태}` 와 같은 우선순위 지정도 가능하다.

YAML

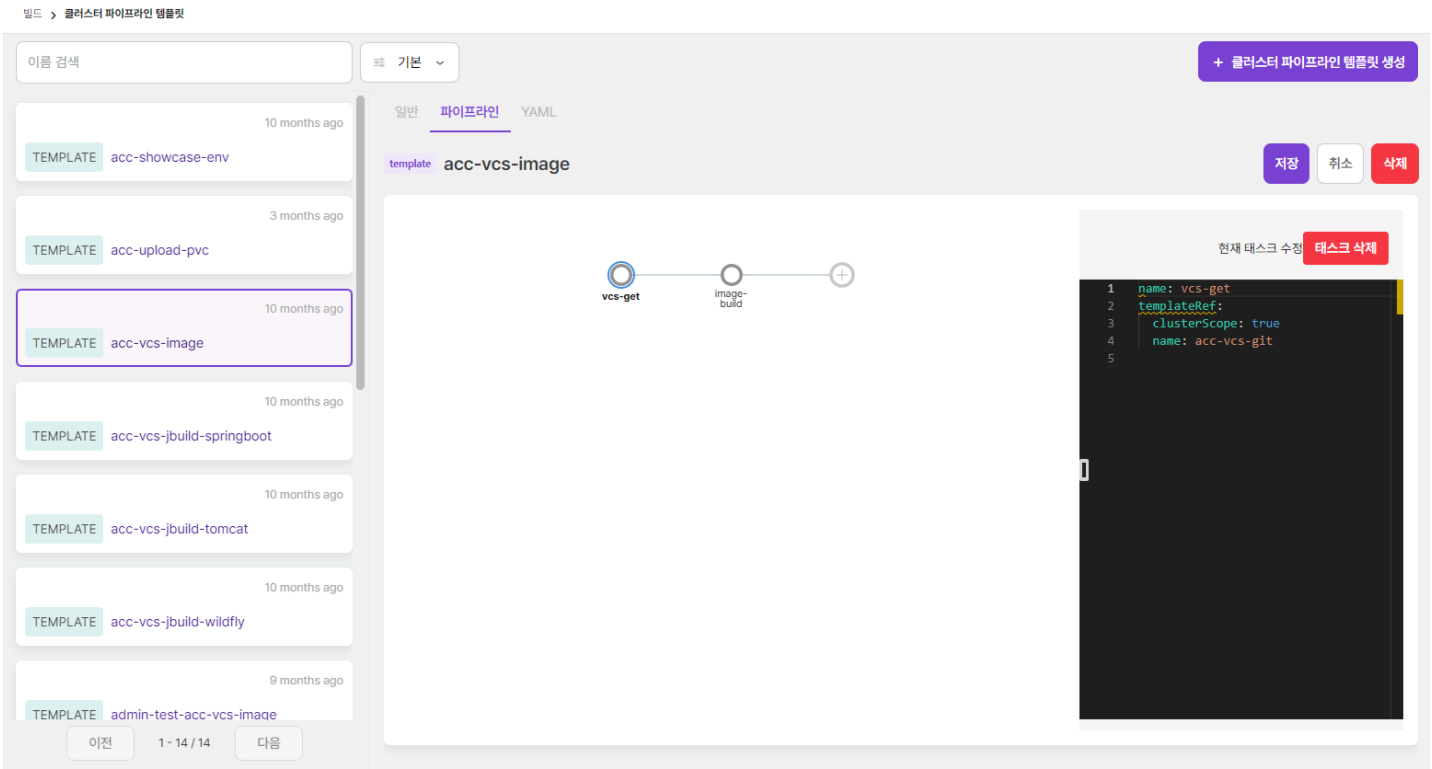
```yaml
spec:
  tasks:
    - name: vcs-get
      templateRef:
        clusterScope: true
        name: acc-vcs-git
    - name: test-task
      templateRef:
        clusterScope: true
        name: test-task-template
    - depends: (test-task.Succeeded || test-task.Running) && vcs-get.Succeeded
     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

## 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.

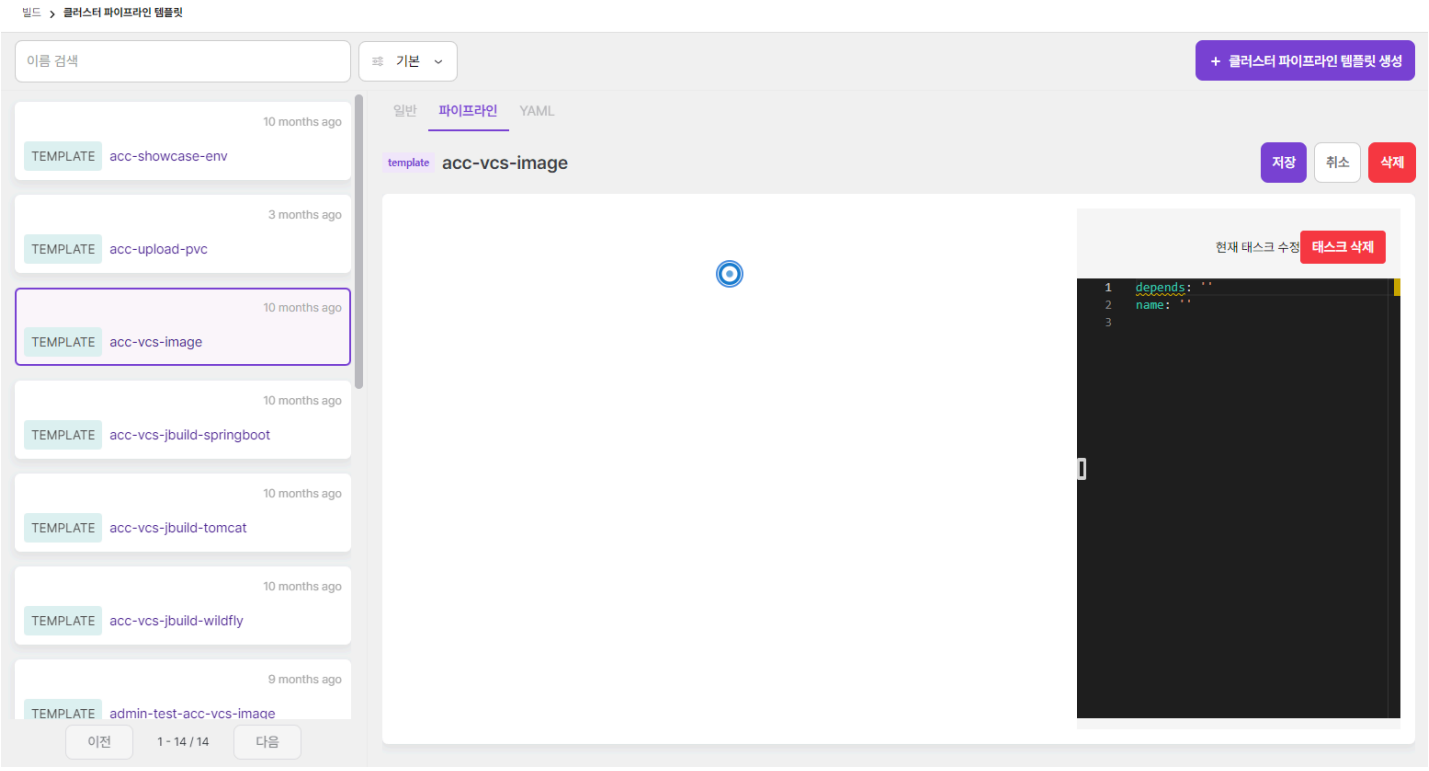



**NOTE**

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

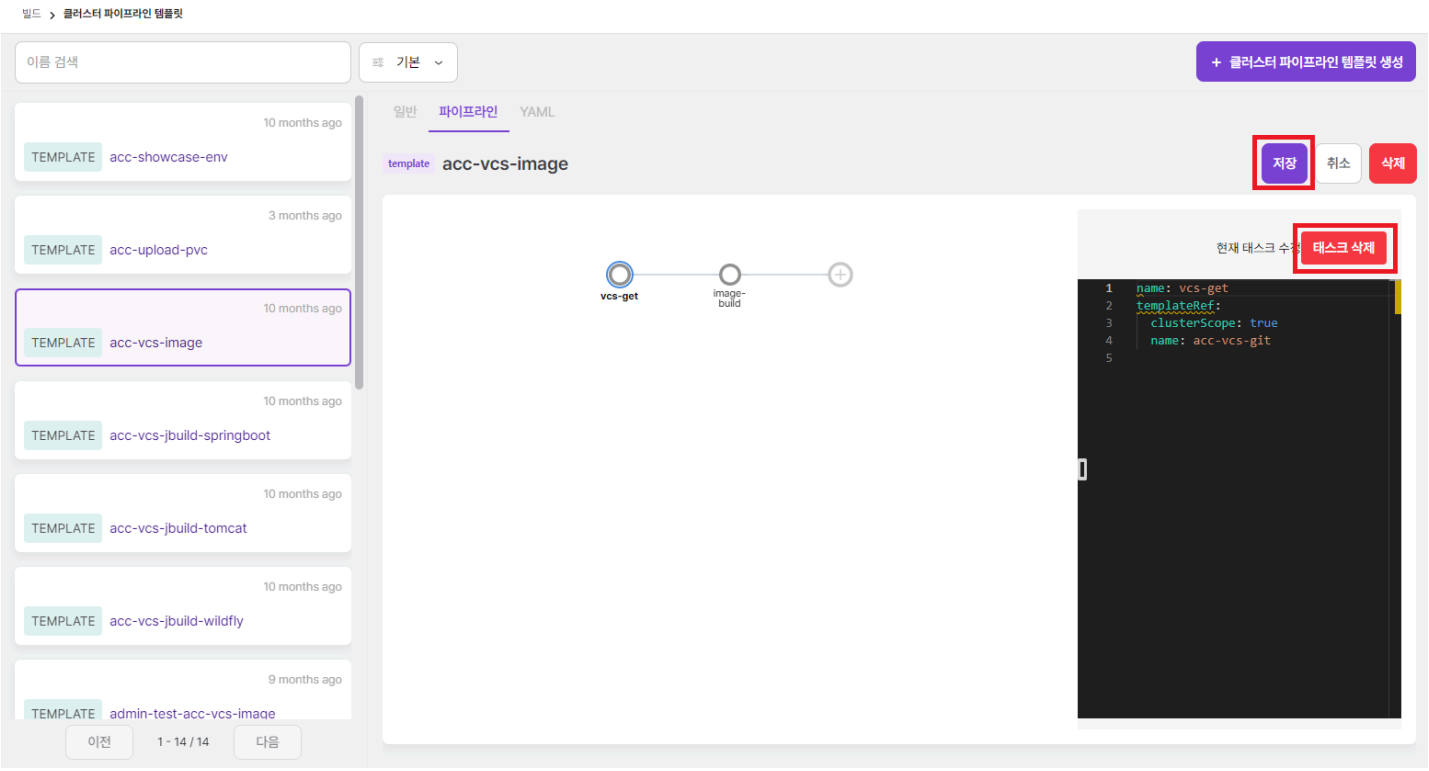
## 태스크 생성

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



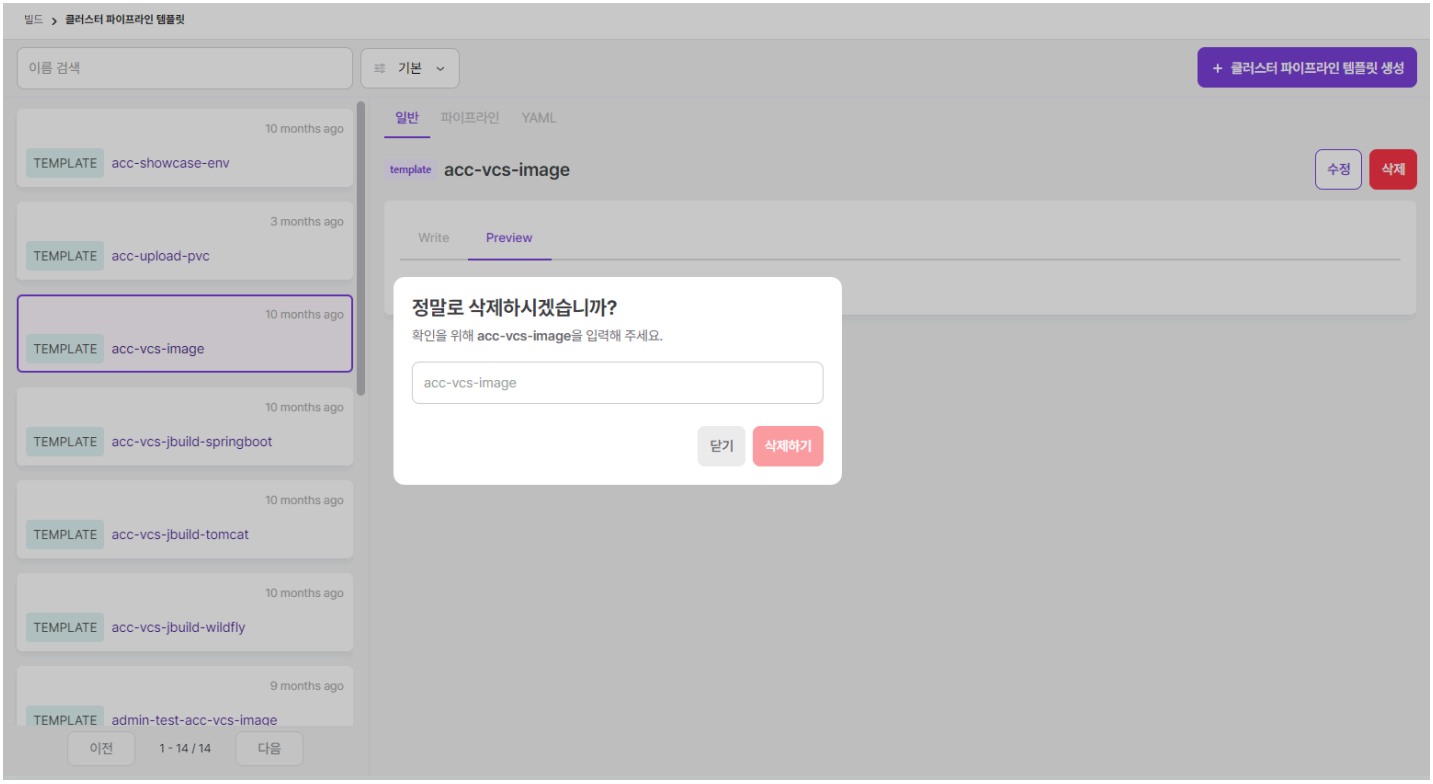

## 태스크 삭제

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.

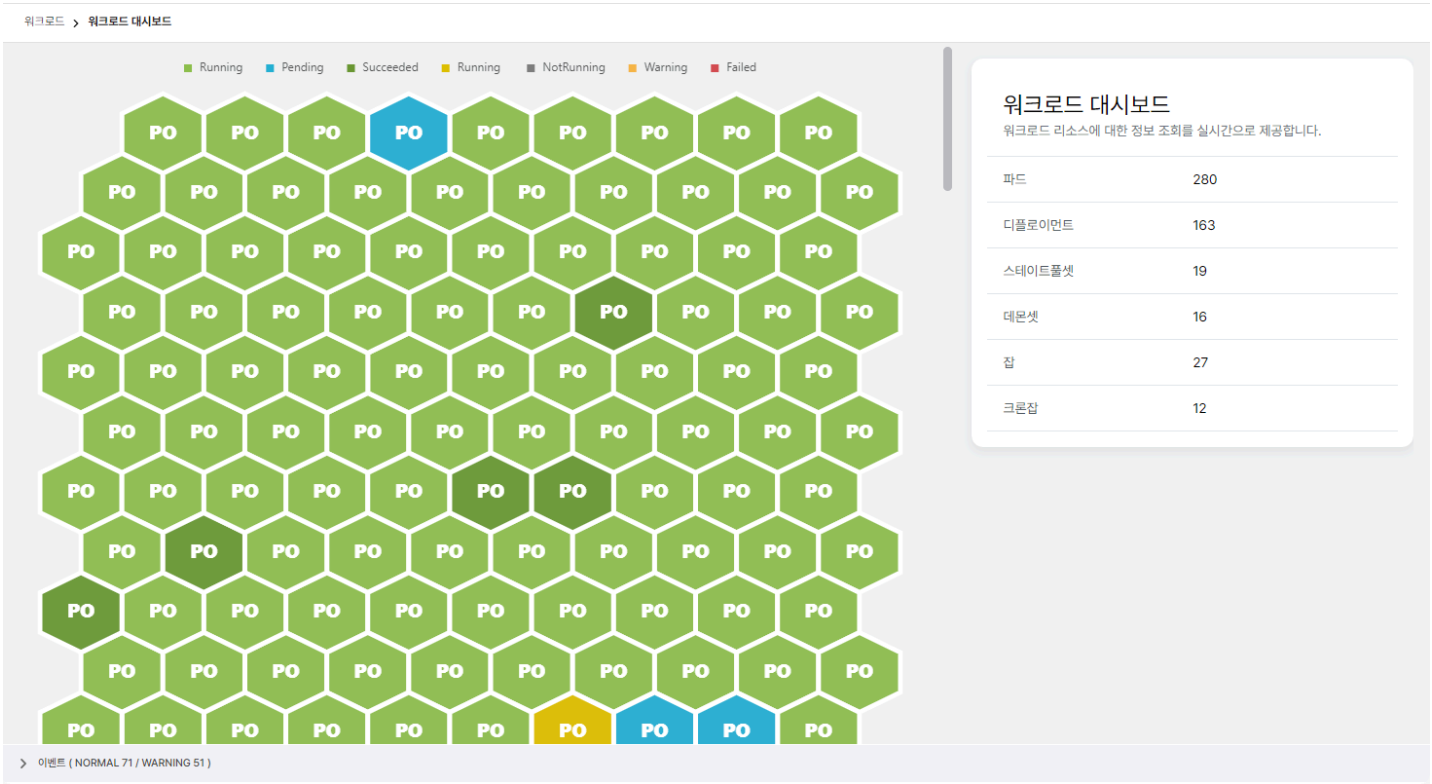
### 4.2.6. 워크로드

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

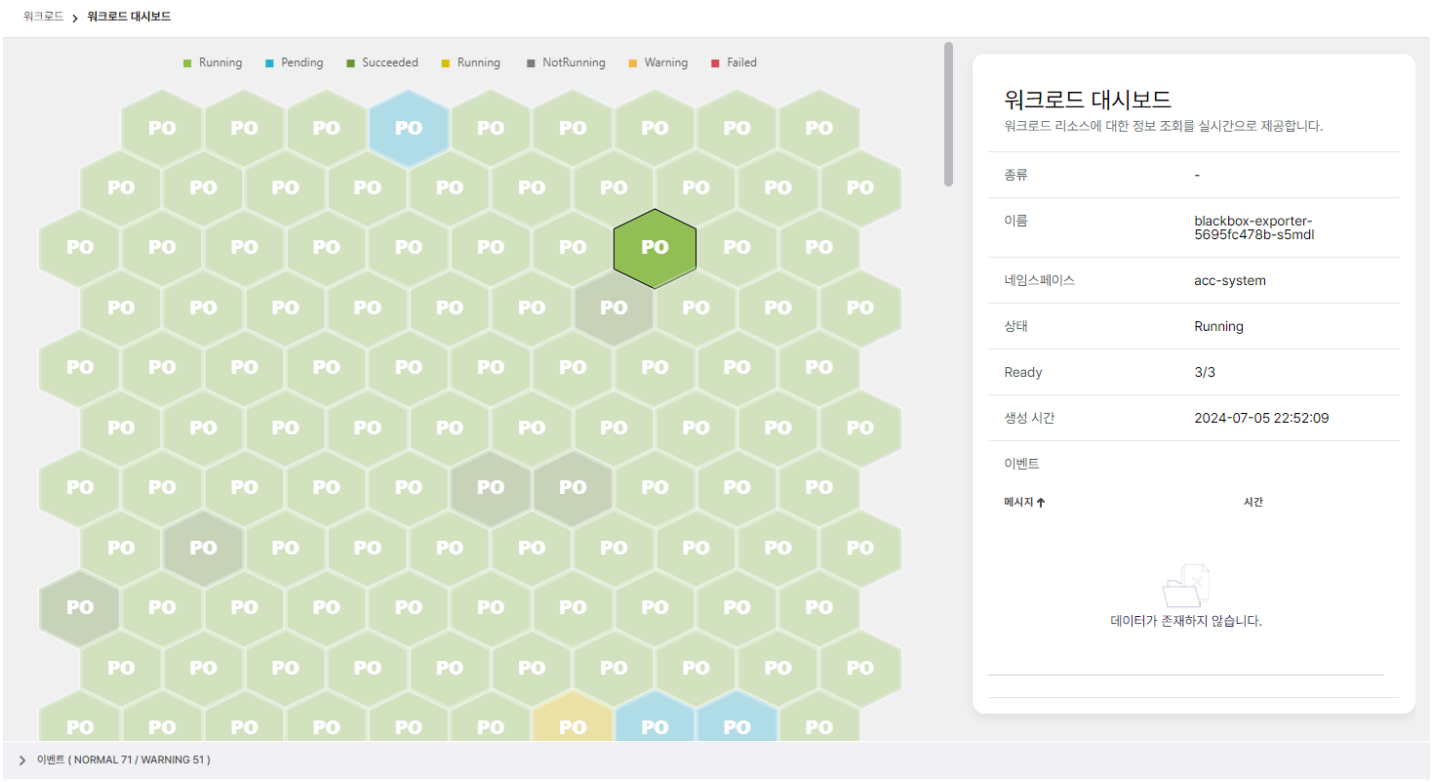
- 디플로이먼트
- 스테이트풀셋
- 데몬셋
- 레플리카셋
- 잡
- 크론잡
- 파드

#### 4.2.6.1. 워크로드 대시보드

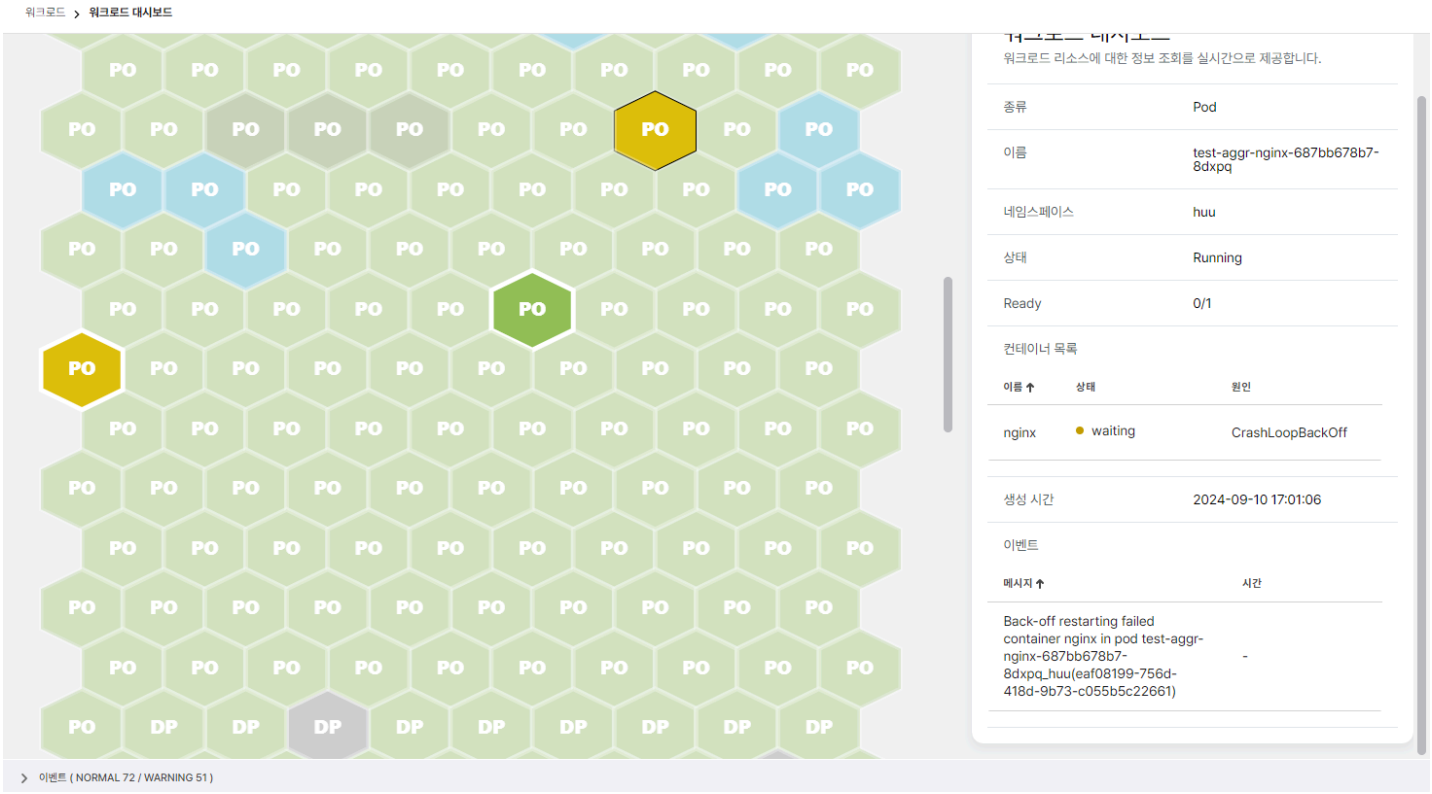
워크로드 대시보드는 클러스터에 배포된 워크로드 상태 정보를 제공한다.



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




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



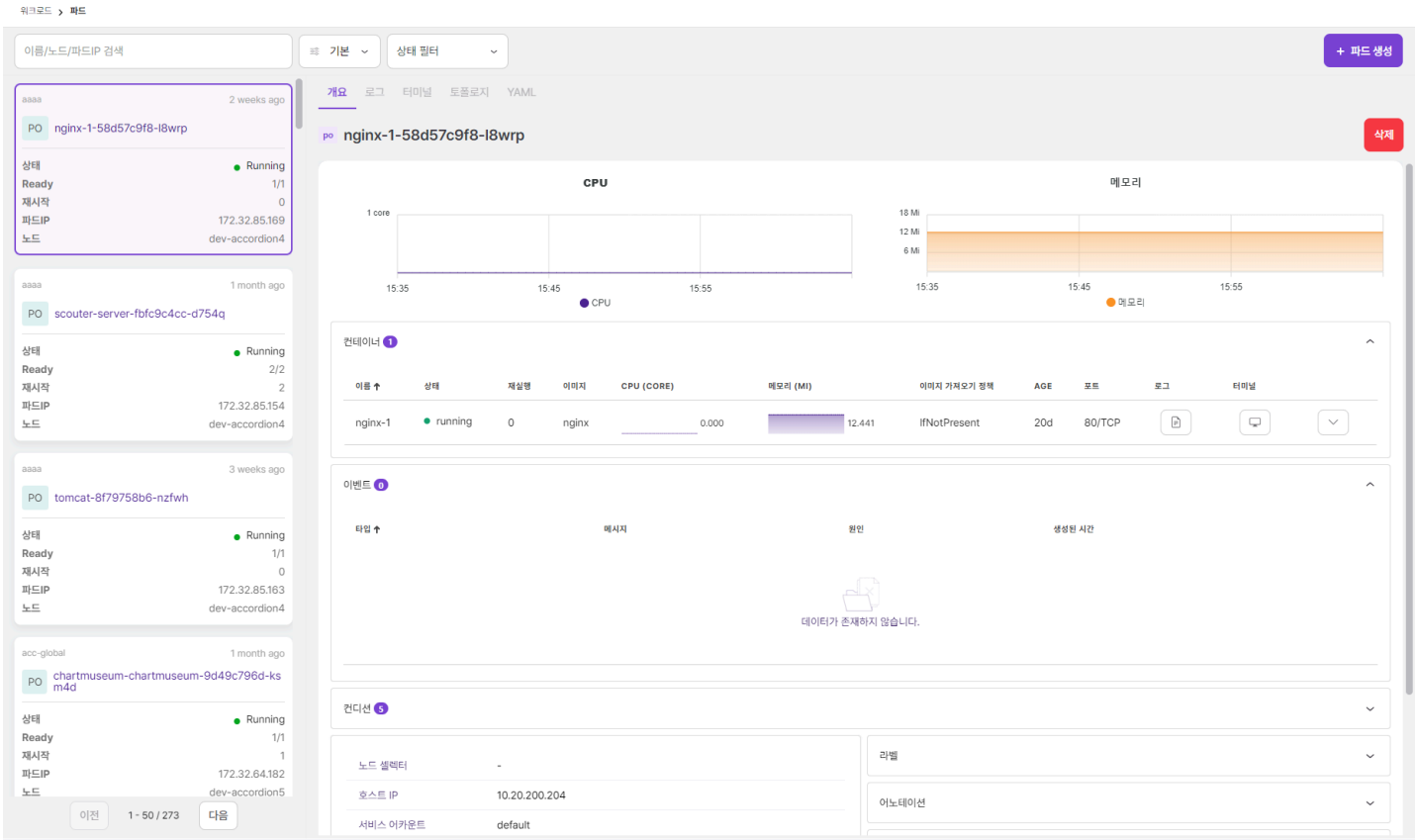
## 4.2.6.2. 파드

배포된 파드 정보를 제공한다.

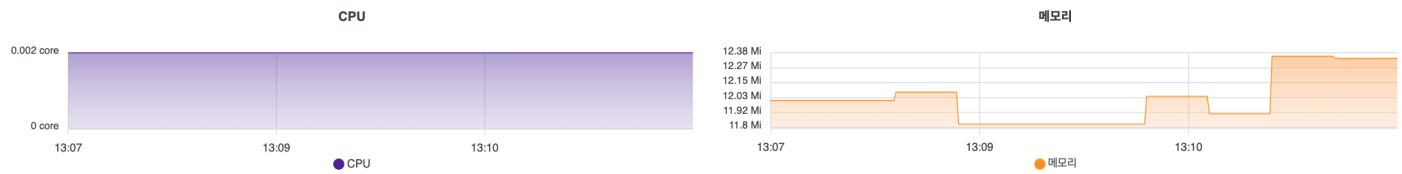
탭을 이용해 파드에 대한 쿠버네티스 리소스 정보, 컨테이너 로그 조회 및 터미널 접속, 토폴로지 조회가 가능하다.
컨테이너 로그 조회와 터미널 접속은 파드의 컨테이너 별로 가능하다.

### 개요

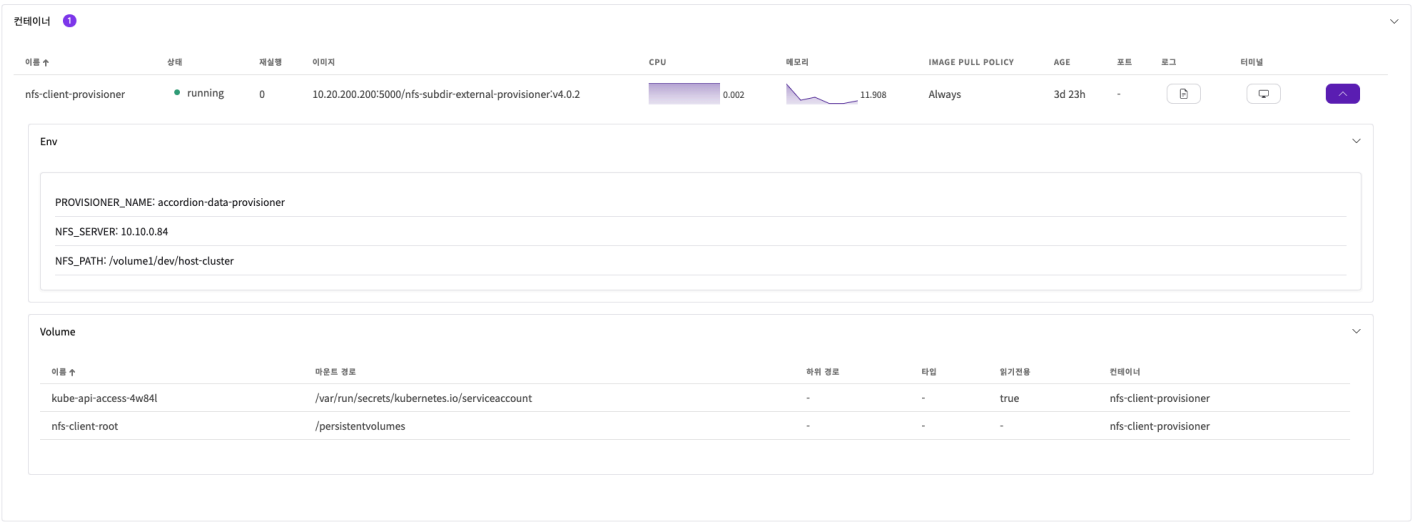
배포된 파드의 메트릭, 컨테이너, 이벤트, 상태, 세부 정보를 제공한다.



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

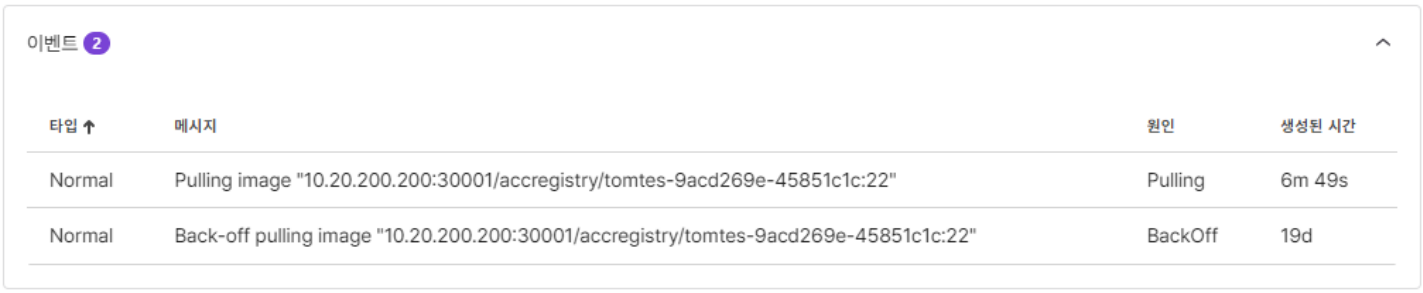



컨테이너 정보
배포된 파드의 컨테이너에 대한 정보를 제공한다. 컨테이너 테이블 열을 선택하면 Arguments, Env, Volume에 대한 정보를 조회할 수 있다.




| 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.




| 항목 | 설명 |
| --- | --- |
| 타입 | 이벤트 타입<br>• Normal: 일반적인 작업으로 발생된 이벤트<br>• Warning: 오류에 의해 발생된 이벤트 |
| 메시지 | 이벤트 메시지 |
| 원인 | 이벤트 발생 이유 |
| 생성된 시간 | 이벤트 생성 후 지난 시간 |

컨디션 정보

| 항목 | 설명 |
| --- | --- |
| 타입 | 파드 컨디션의 이름<br>• PodReadyToStartContainers: 파드가 컨테이너를 시작할 준비가 되었다.<br>• PodScheduled: 파드가 노드에 스케줄되었다.<br>• ContainersReady: 파드의 모든 컨테이너가 준비되었다.<br>• Initialized: 모든 초기화 컨테이너가 성공적으로 완료(completed)되었다.<br>• 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 |

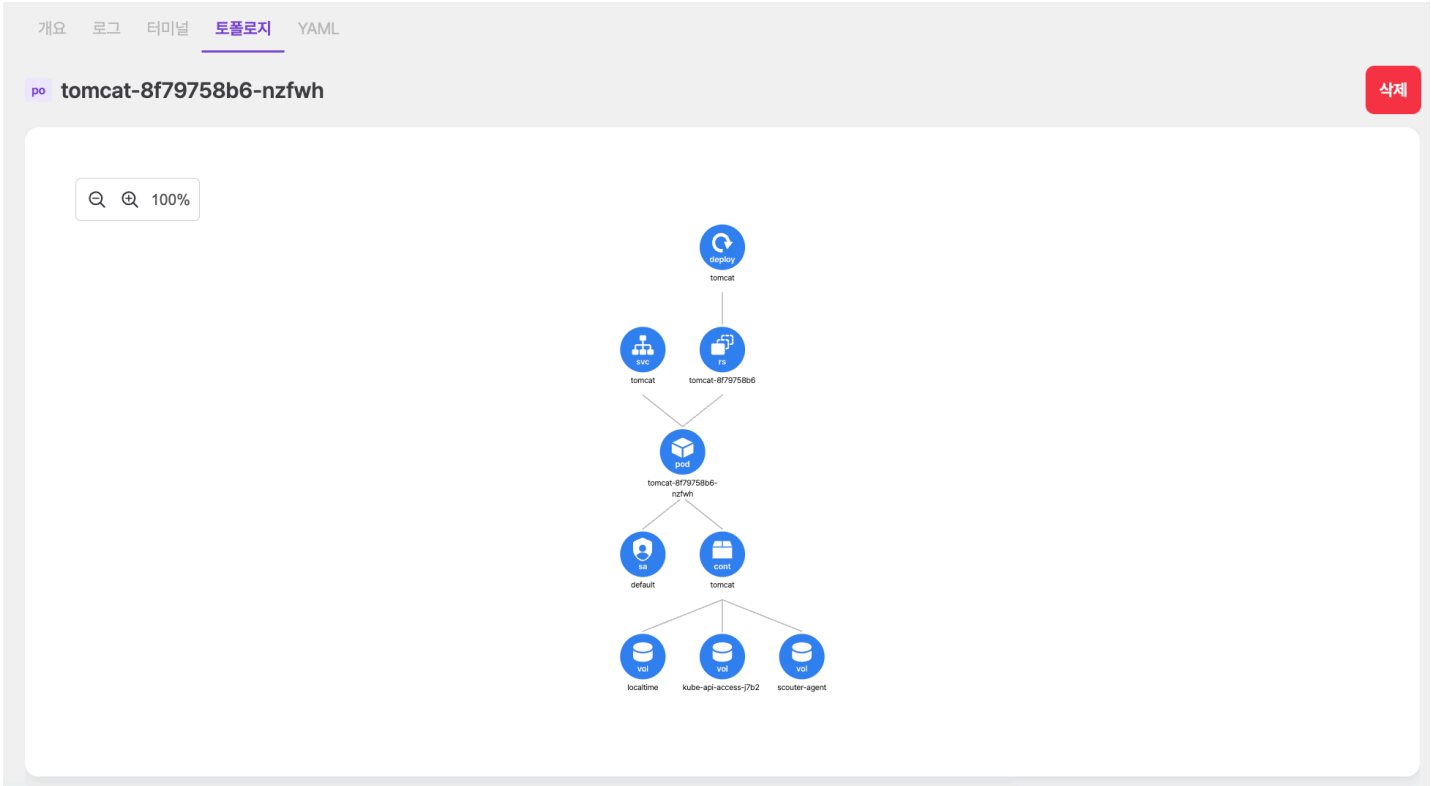
파드 세부 정보
파드에 대한 세부 정보를 제공한다.

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

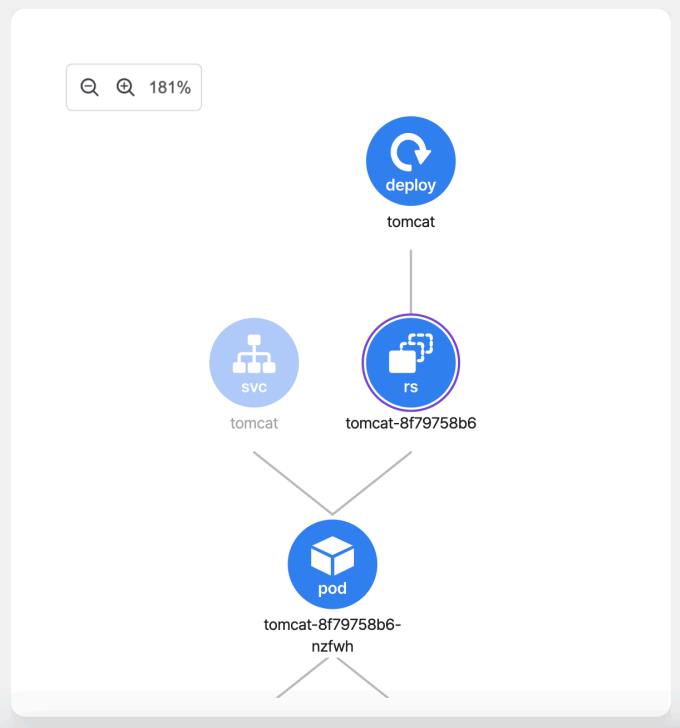
라벨 >
어노테이션 >
톨러레이션 >
어피니티 >

토폴로지
파드와 관련된 리소스를 토폴로지 형태로 보여준다.
파드의 토폴로지에서 출력되는 리소스는 다음과 같다.

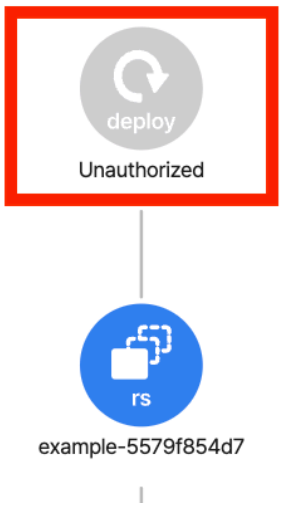
- 파드, 디플로이먼트, 레플리카셋, 스테이트풀셋, 데몬셋, 잡
- 서비스
- 서비스어카운트
- 퍼시스턴트볼륨클레임
- 시크릿, 컨피그맵
- 컨테이너, 볼륨 (해당 리소스들은 단순 연관 관계를 출력하는 리소스로 클릭하여도 화면은 이동하지 않는다.)




리소스에 마우스 오버 시, 직접적으로 연관이 있는 리소스만 활성화되어 보여진다.
리소스 클릭 시에는 해당 리소스의 화면으로 이동한다.

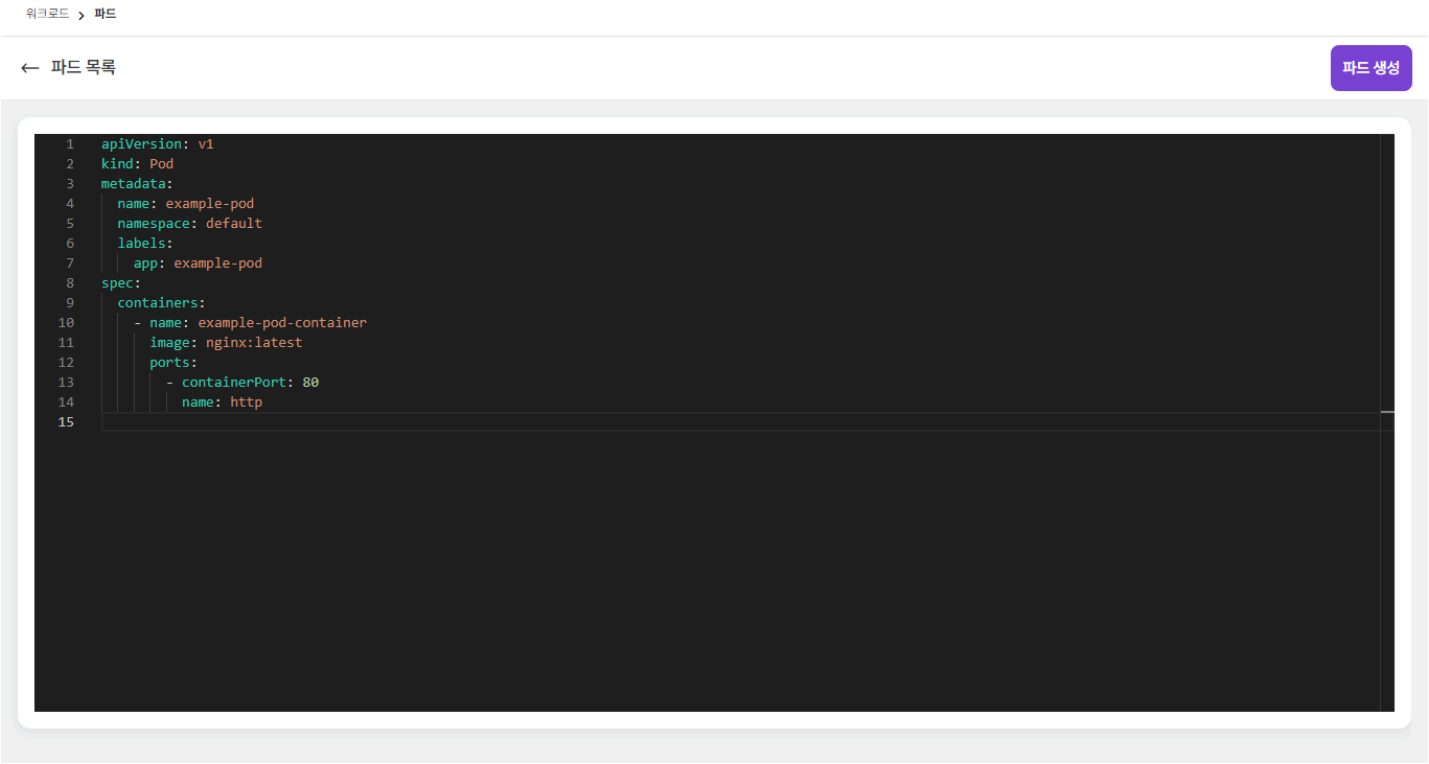



만약 권한이 없는 리소스가 연관관계에 있을 경우에는, 해당 이미지와 같이 리소스의 이름 없이 비활성화된 노드 형태로 리소스의 종류만 출력하며 클릭하여도 화면 이동이 일어나지 않는다.



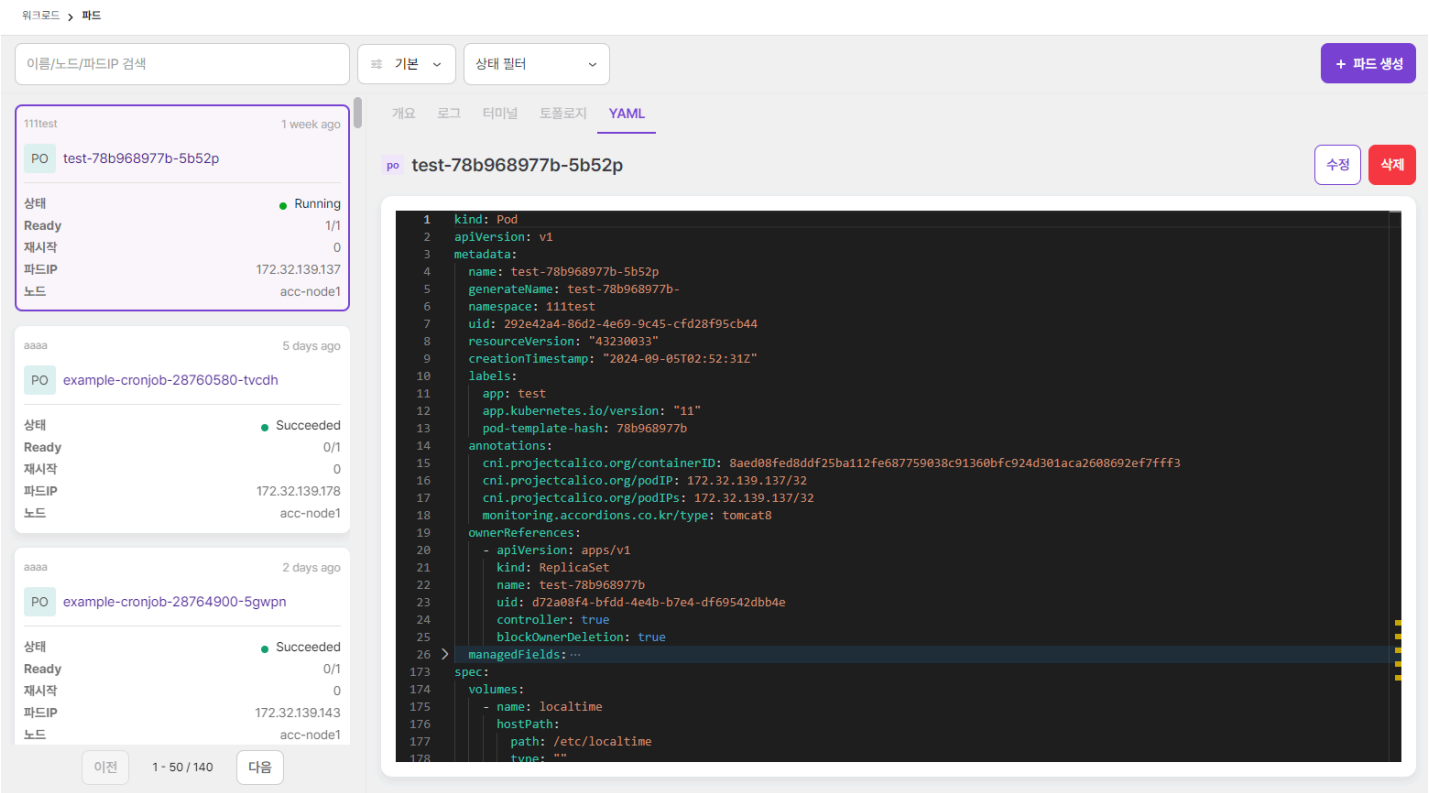
## 파드 생성

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



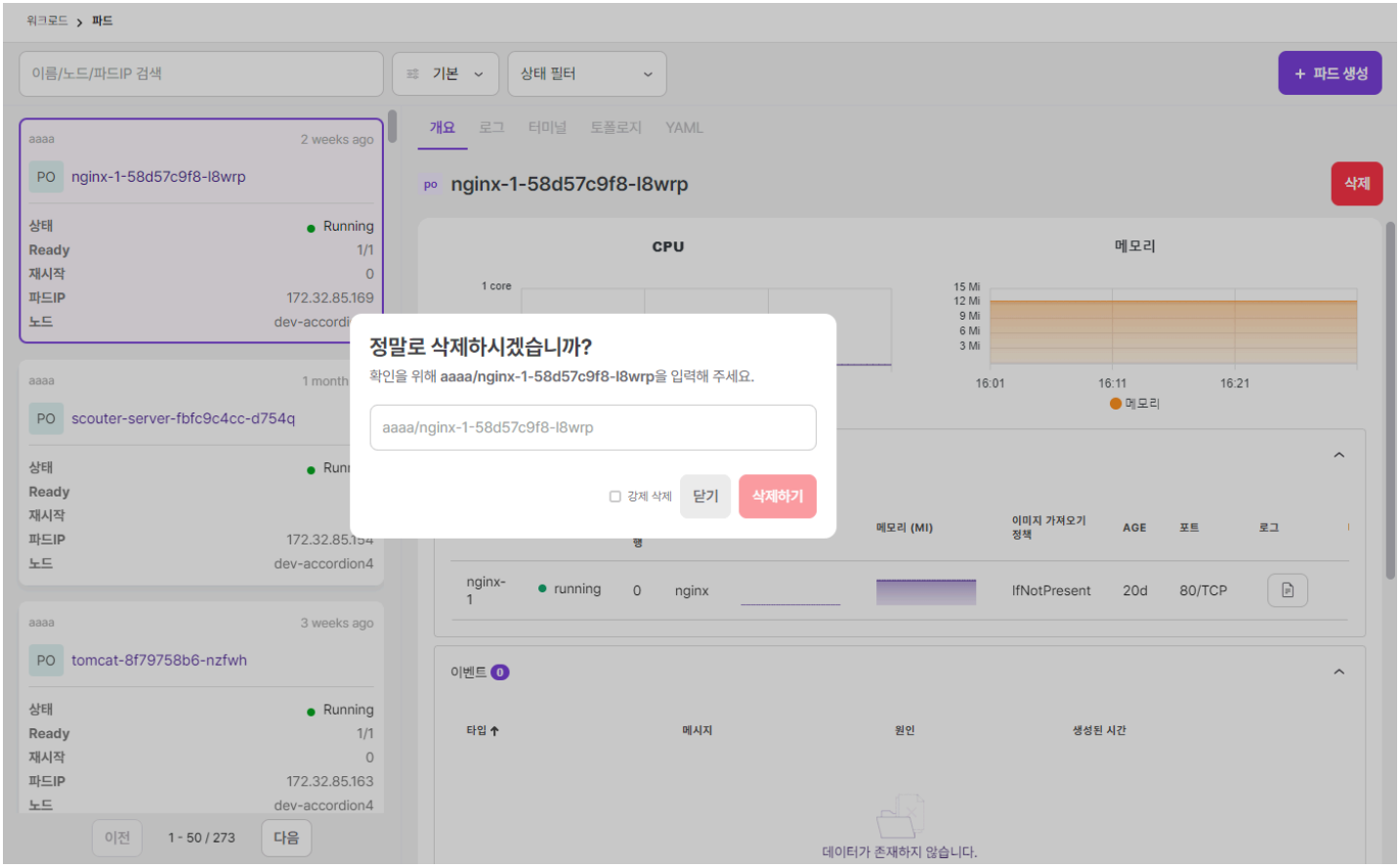

## 파드 수정

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



## 파드 삭제

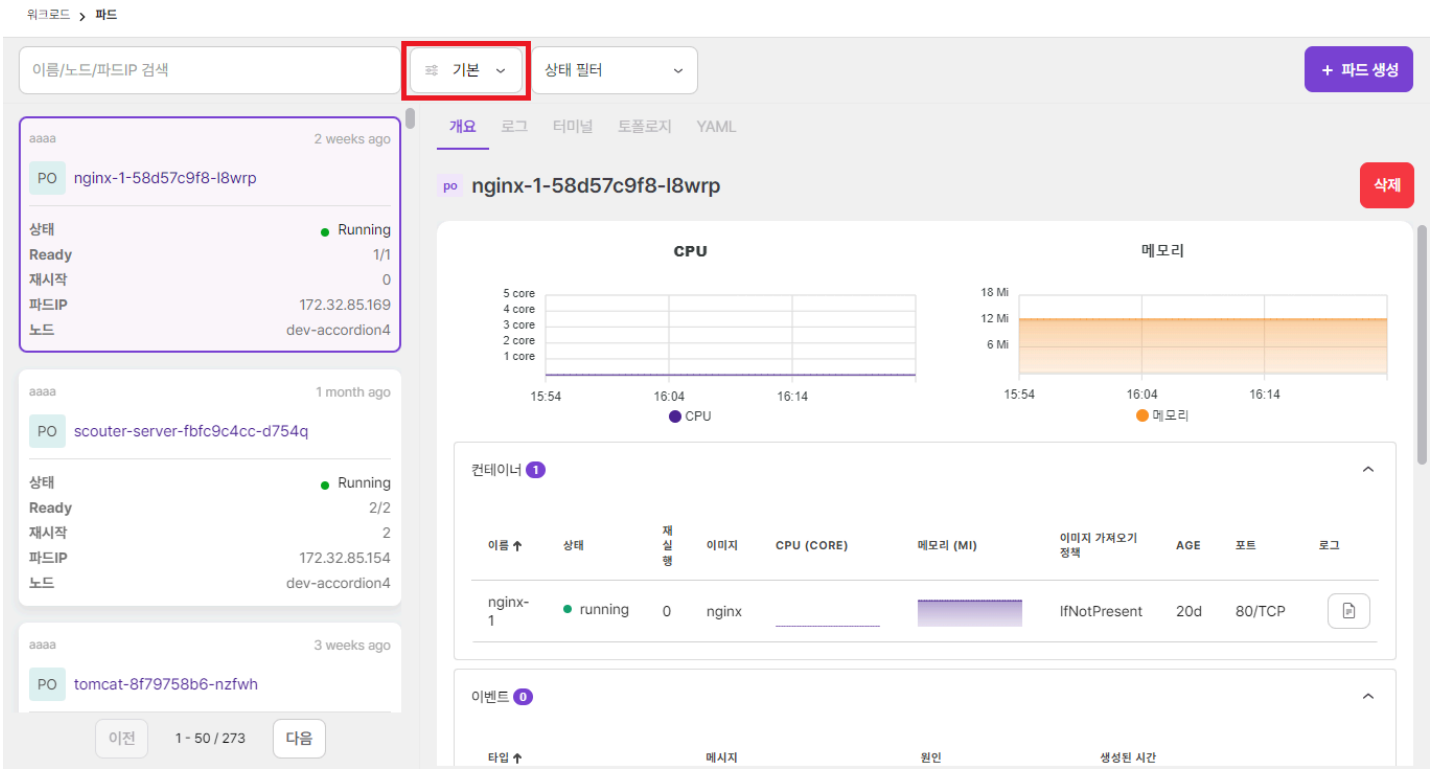
삭제하려는 파드를 선택하고 우측의 삭제 버튼을 선택한다.

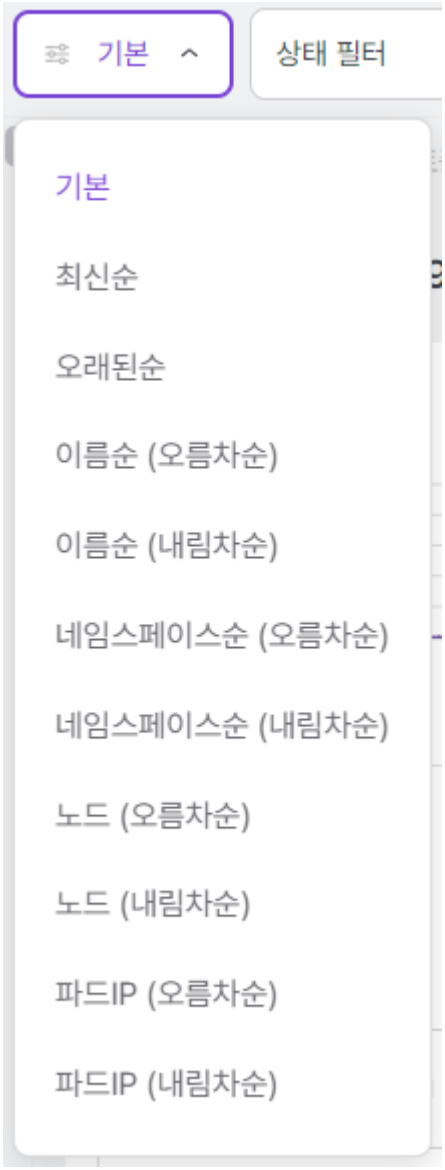



모달에서 네임스페이스와 파드 이름을 입력하여 삭제한다.

## 파드 정렬

파드의 정렬을 지정하여 조회한다.

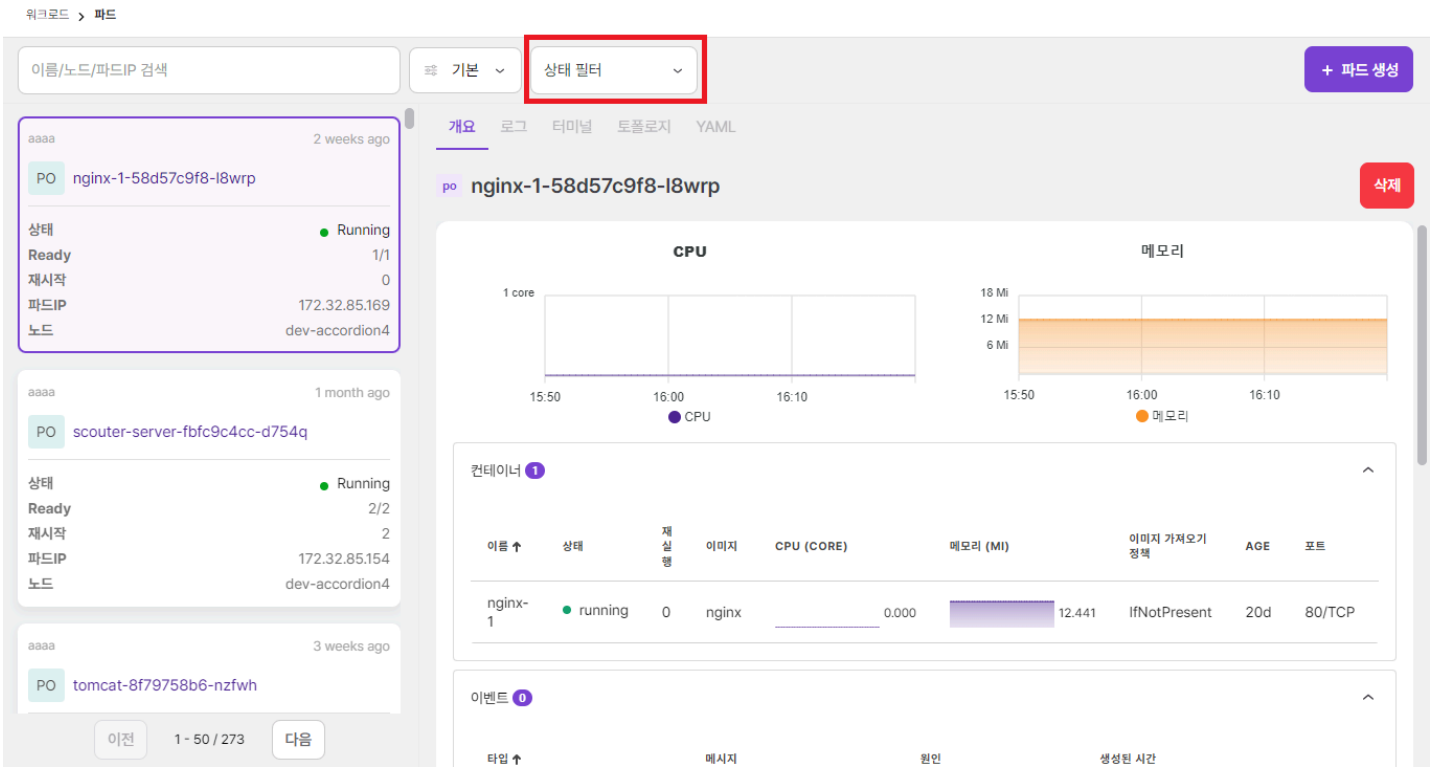


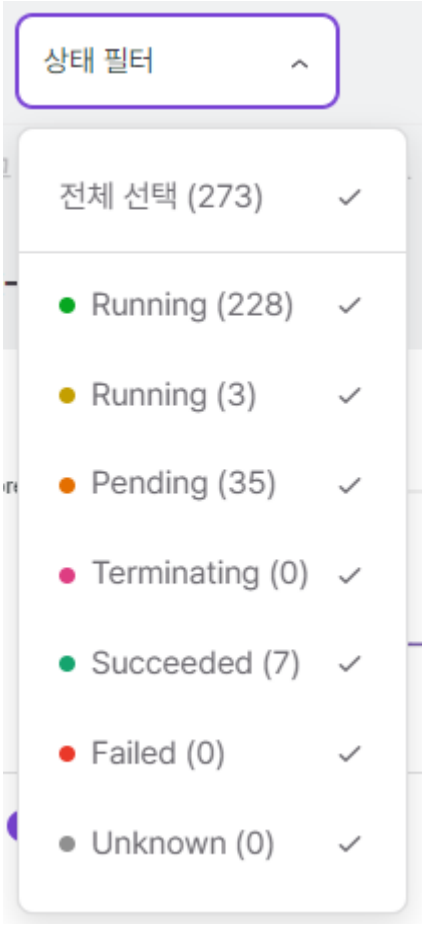



| 이름 | 설명 |
|---|---|
| 최신순 | 파드의 실행된 시간이 최신인 순으로 파드를 조회한다. |
| 오래된순 | 파드의 실행된 시간이 오래된순으로 파드를 조회한다. |
| 이름순 (오름차순) | 파드 이름의 오름차순으로 파드를 조회한다. |
| 이름순 (내림차순) | 파드 이름의 내림차순으로 파드를 조회한다. |
| 네임스페이스순 (오름차순) | 네임스페이스 이름의 오름차순으로 파드를 조회한다. |
| 네임스페이스순 (내림차순) | 네임스페이스 이름의 내림차순으로 파드를 조회한다. |
| 노드 (오름차순) | 파드가 속한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

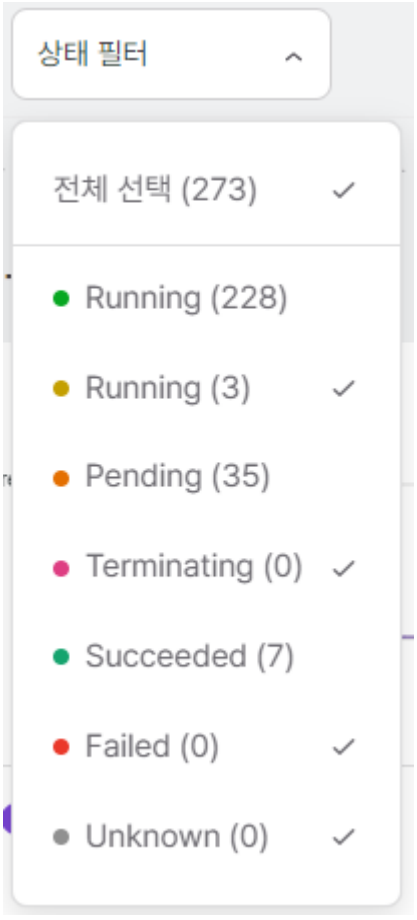


| 항목 | 설명 |
| --- | --- |
| Running | 동작 중인 파드를 의미한다. |
| Running | 동작 중이지만 비정상적인 컨테이너가 존재하는 파드를 의미한다. |
| Pending | 실행할 준비가 되지 않은 파드를 의미한다. |
| Terminating | 종료중인 파드를 의미한다. |
| Succeeded | 성공적으로 종료된 파드를 의미한다. |
| Failed | 내부 컨테이너가 실패로 종료된 파드를 의미한다. |
| Unknown | 파드의 상태를 얻을 수 없는 경우를 의미한다. |

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

필터링 상태는 다중 선택이 가능하다. 기본값은 `전체 선택` 이다.

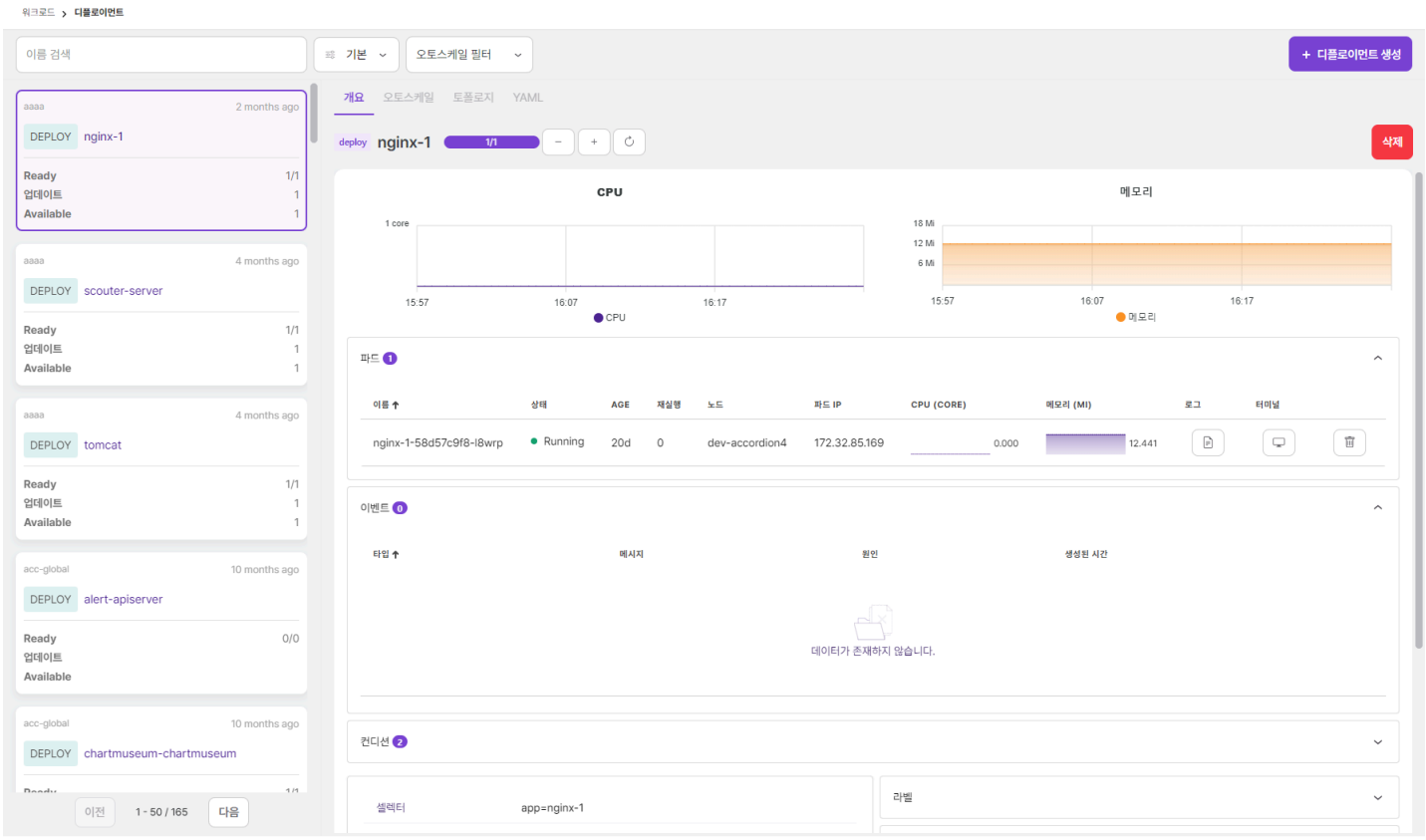


## 4.2.6.3. 디플로이먼트

배포된 디플로이먼트 정보를 제공한다.

### 개요

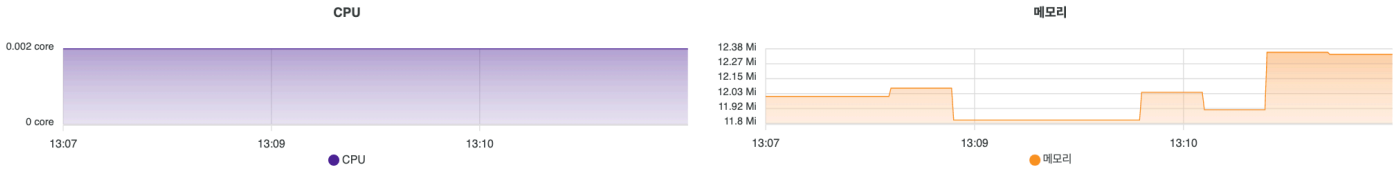
배포된 디플로이먼트의 메트릭, 파드, 이벤트, 컨디션, 세부 정보를 제공한다.



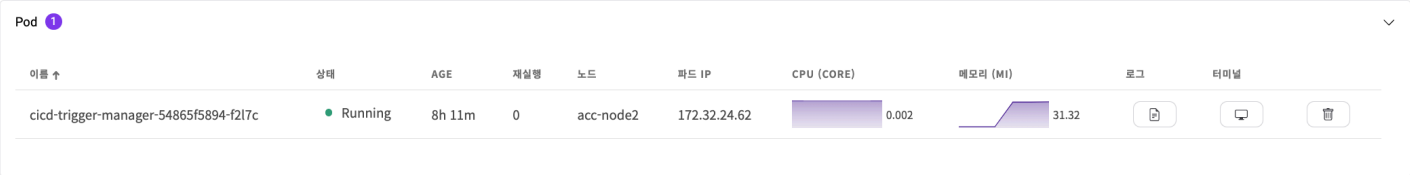

스케일 및 롤아웃

| 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.

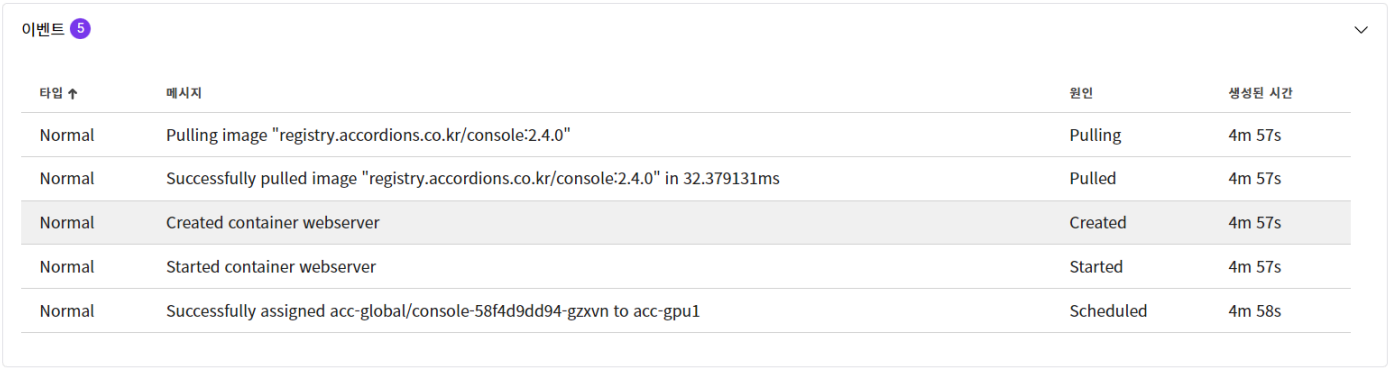


파드 정보
배포된 파드에 대한 정보를 제공한다. 클릭하면 해당 파드 화면으로 이동한다.




| 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 |

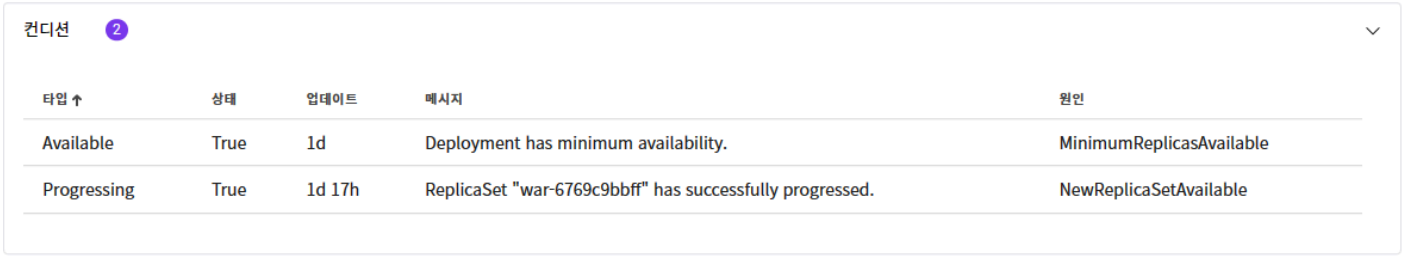
이벤트 정보
파드에서 발생하는 이벤트 정보를 제공한다.




| 항목 | 설명 |
| --- | --- |
| 타입 | 이벤트 타입<br>• Normal: 일반적인 작업으로 발생된 이벤트<br>• Warning: 오류에 의해 발생된 이벤트 |

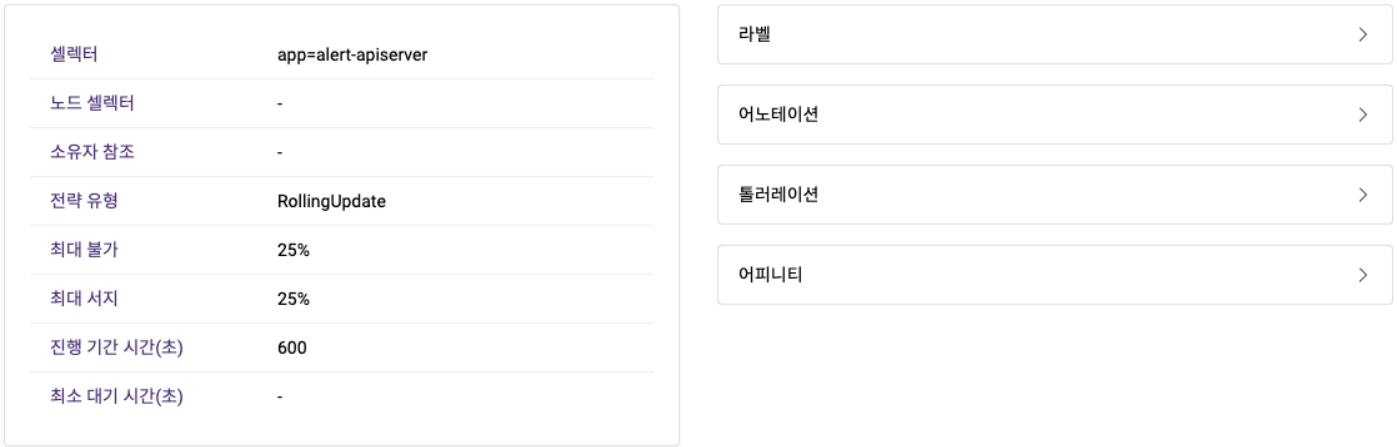
| 항목 | 설명 |
| --- | --- |
| 메시지 | 이벤트 메시지 |
| 원인 | 이벤트 발생 이유 |
| 생성된 시간 | 이벤트 생성 후 지난 시간 |

컨디션 정보




| 항목 | 설명 |
| --- | --- |
| 타입 | 디플로이먼트 컨디션의 이름<br>• Available: 디플로이먼트가 사용 가능하다. (기준: 디플로이먼트의 리플리카 값)<br>• Progressing: 리플리카셋에 대한 배포 진행 상황을 표시한다.<br>• 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 |

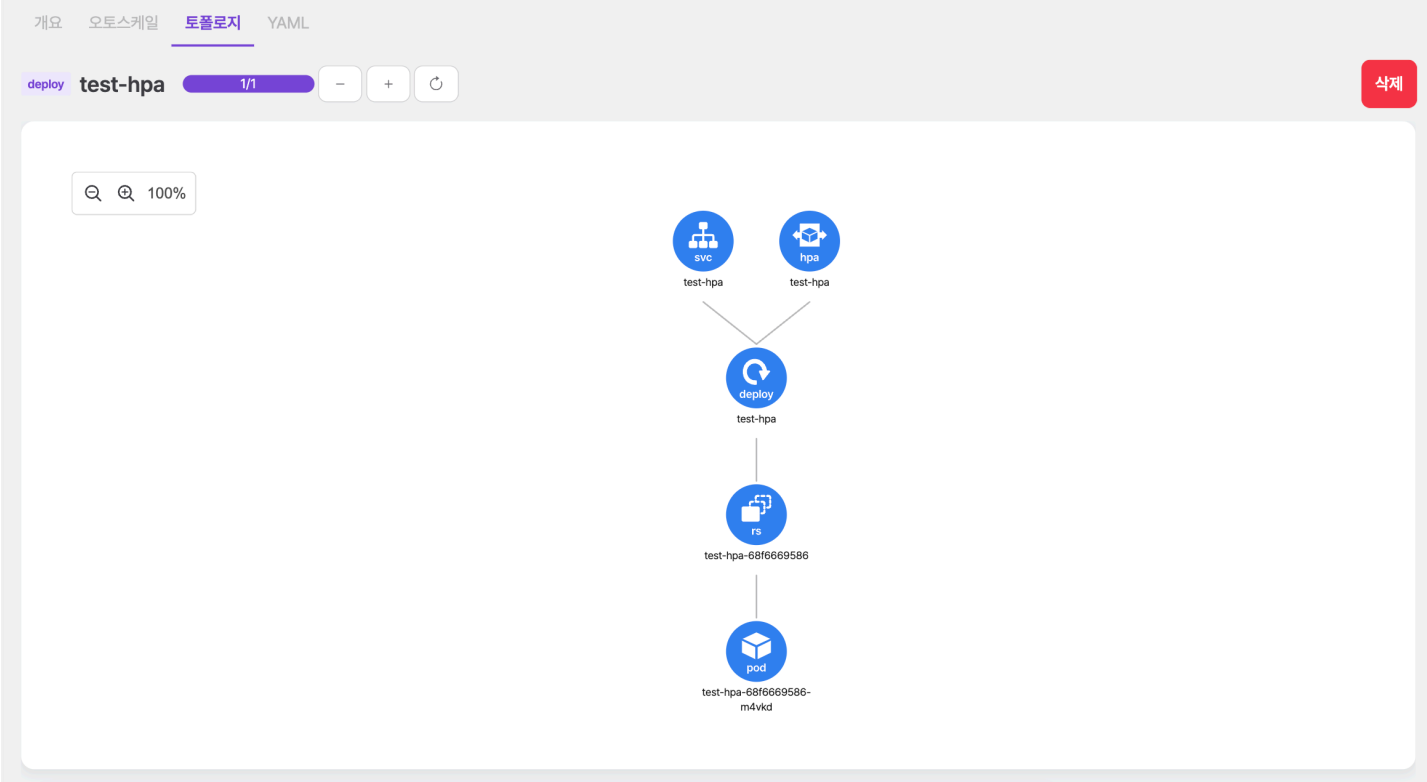
세부 정보
디플로이먼트에 대한 세부 정보를 제공한다.



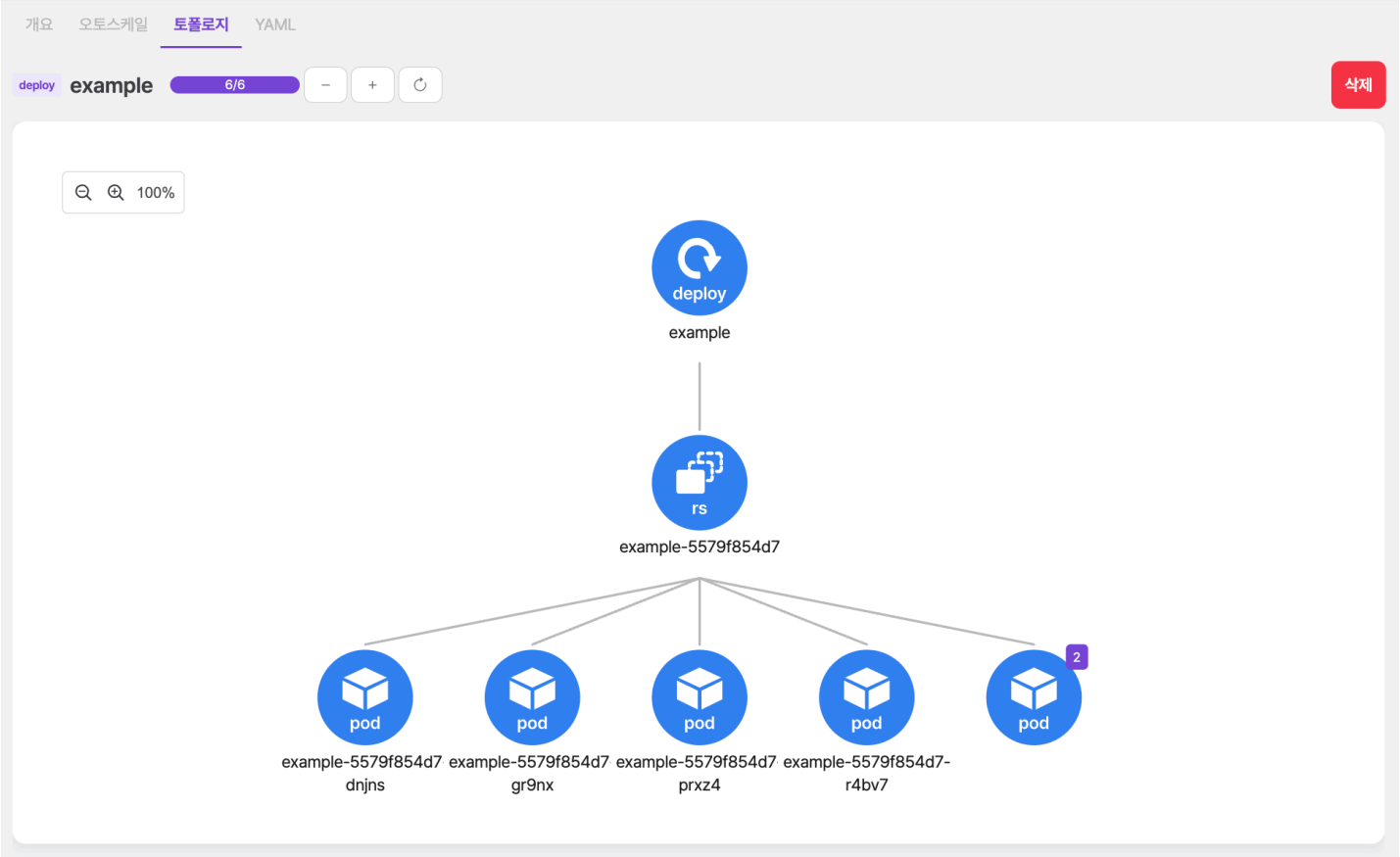
토폴로지
디플로이먼트와 관련된 리소스를 토폴로지 형태로 보여준다.
기본 기능은 파드와 내용이 동일하기 때문에 파드 토폴로지를 참고한다.

디플로이먼트의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 디플로이먼트, 레플리카셋, 파드
- HPA
- 서비스



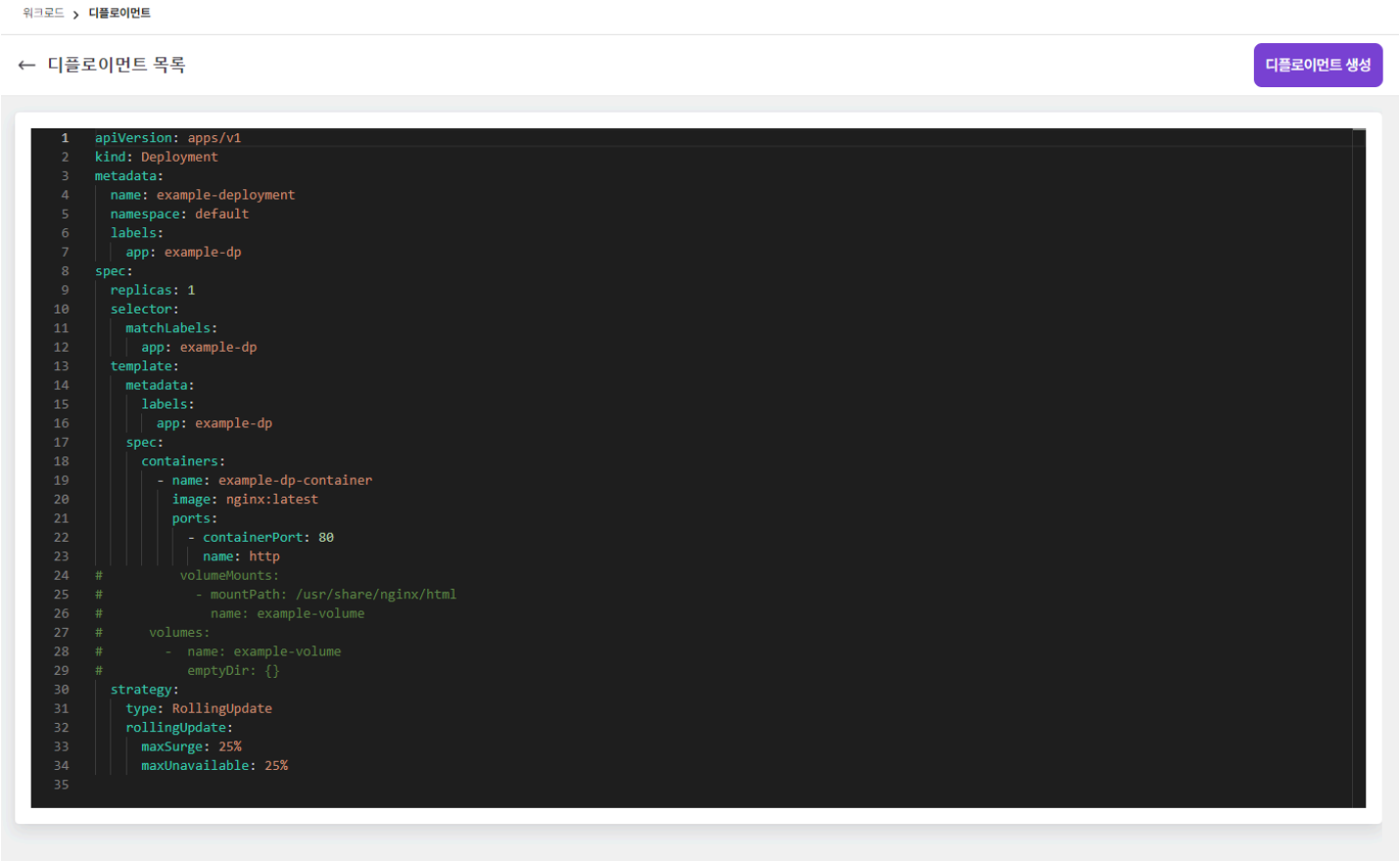
디플로이먼트 토폴로지에서 가로로 출력되는 최대 파드 개수는 5개로 파드의 개수가 6개 이상일 경우, 다음과 같이 나머지 파드의 개수를 출력하는 노드로 보여진다.




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.




## 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 |
| --- | --- |
| 특정 시간에 대해 파드의 개수를 스케일링 | 메트릭 값에 따라 파드의 개수를 스케일링 |
| • 시작 Schedule: 스케일링 시작 시간 설정<br>• 시작 Target Pods: 스케일링 시작 시 목표 파드 개수<br>• 종료 Schedule: 스케일링 종료 시간 설정<br>• 종료 Target Pods: 스케일링 종료 시 목표 파드 개수<br>**NOTE** 시작 Target Pods는 종료 Target Pods보다 큰 값이어야 한다. | • Min Pods: 최소 파드의 개수<br>• Max Pods: 스케일링 시 최대 파드의 개수<br>• 오토스케일 기준: 스케일링 기준이 되는 메트릭<br>• 설정값: 스케일링이 발생하는 임계치<br>**NOTE** Max Pods는 Min Pods보다 큰 값이어야 한다. |

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.

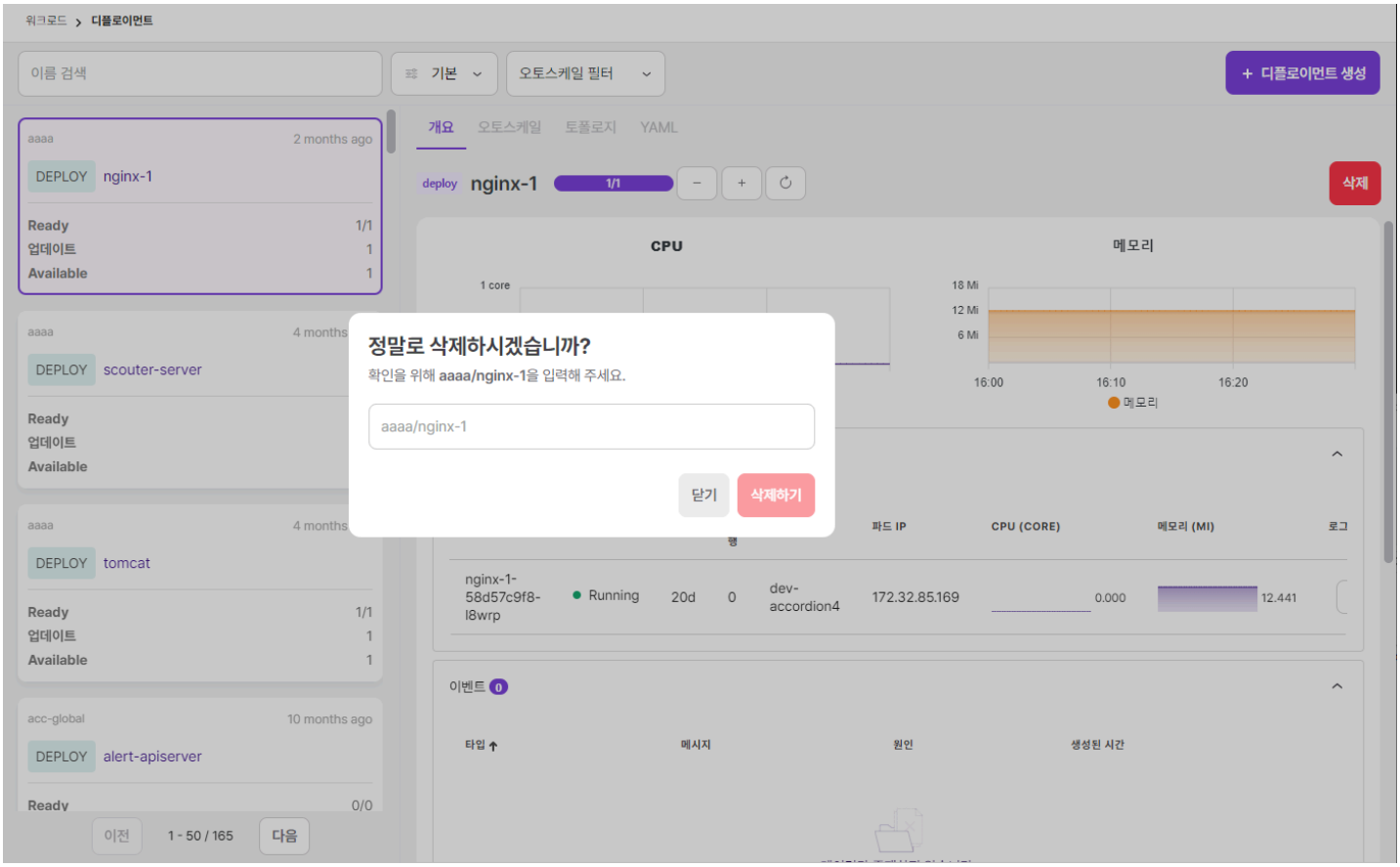
- apm_ActiveService
- apm_TPS
- apm_GcTime
- apm_ElapsedTime

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.



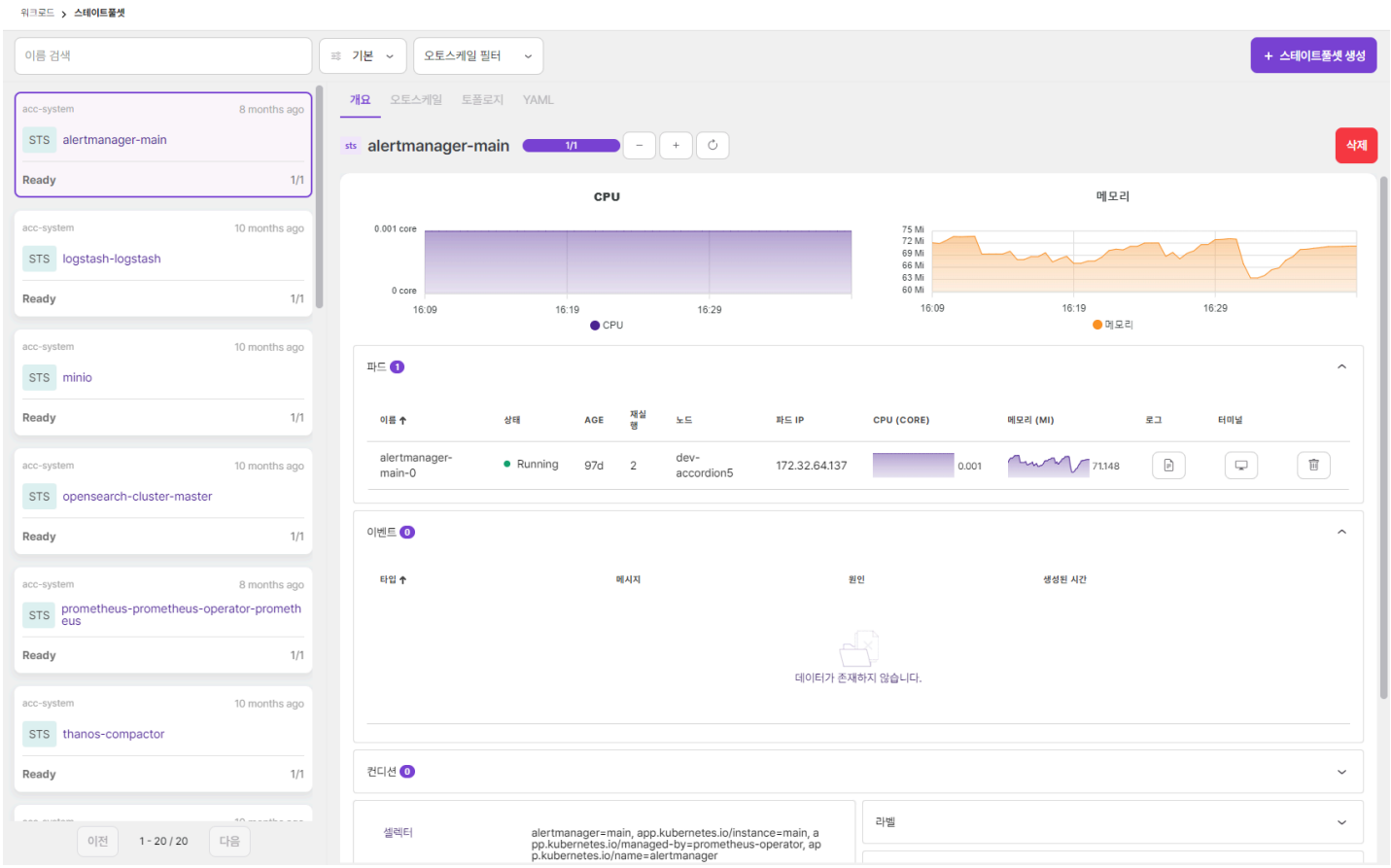

모달에서 네임스페이스와 디플로이먼트 이름을 입력하여 삭제한다.

## 4.2.6.4. 스테이트풀셋

배포된 스테이트풀셋 정보를 제공한다.

### 개요

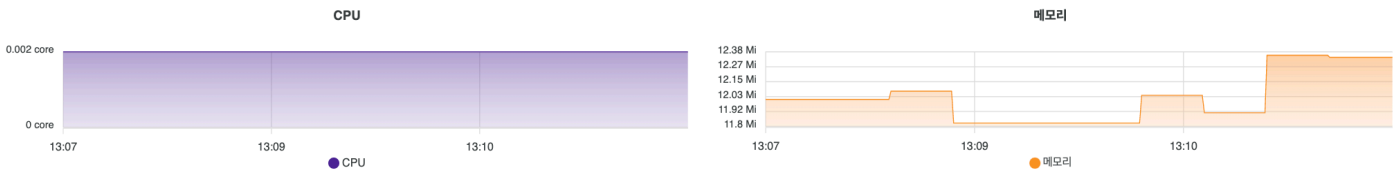
배포된 스테이트풀셋의 메트릭, 파드, 이벤트, 상태, 세부 정보를 제공한다.




스케일 및 롤아웃

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

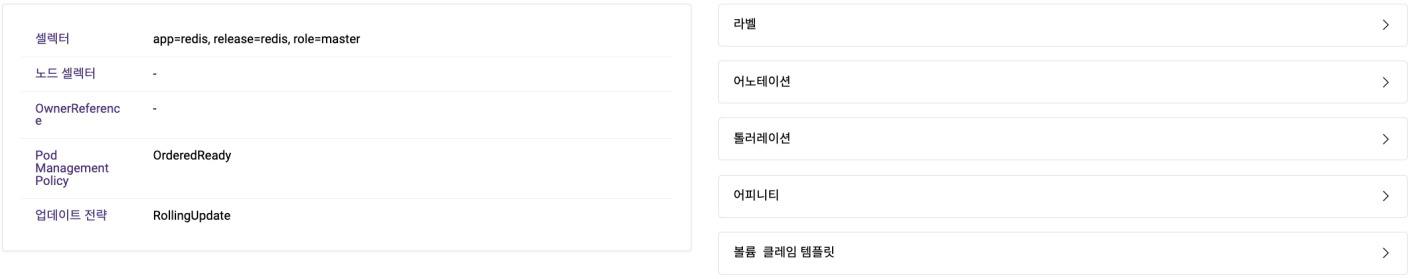
메트릭 정보
메트릭 정보는 현재 시간으로부터 5분간 CPU, Memory 사용량을 제공한다.



파드 정보
디플로이먼트와 내용이 동일하기 때문에 디플로이먼트 파드 정보를 참고한다.

이벤트 정보
파드와 내용이 동일하기 때문에 파드 이벤트 정보를 참고한다.

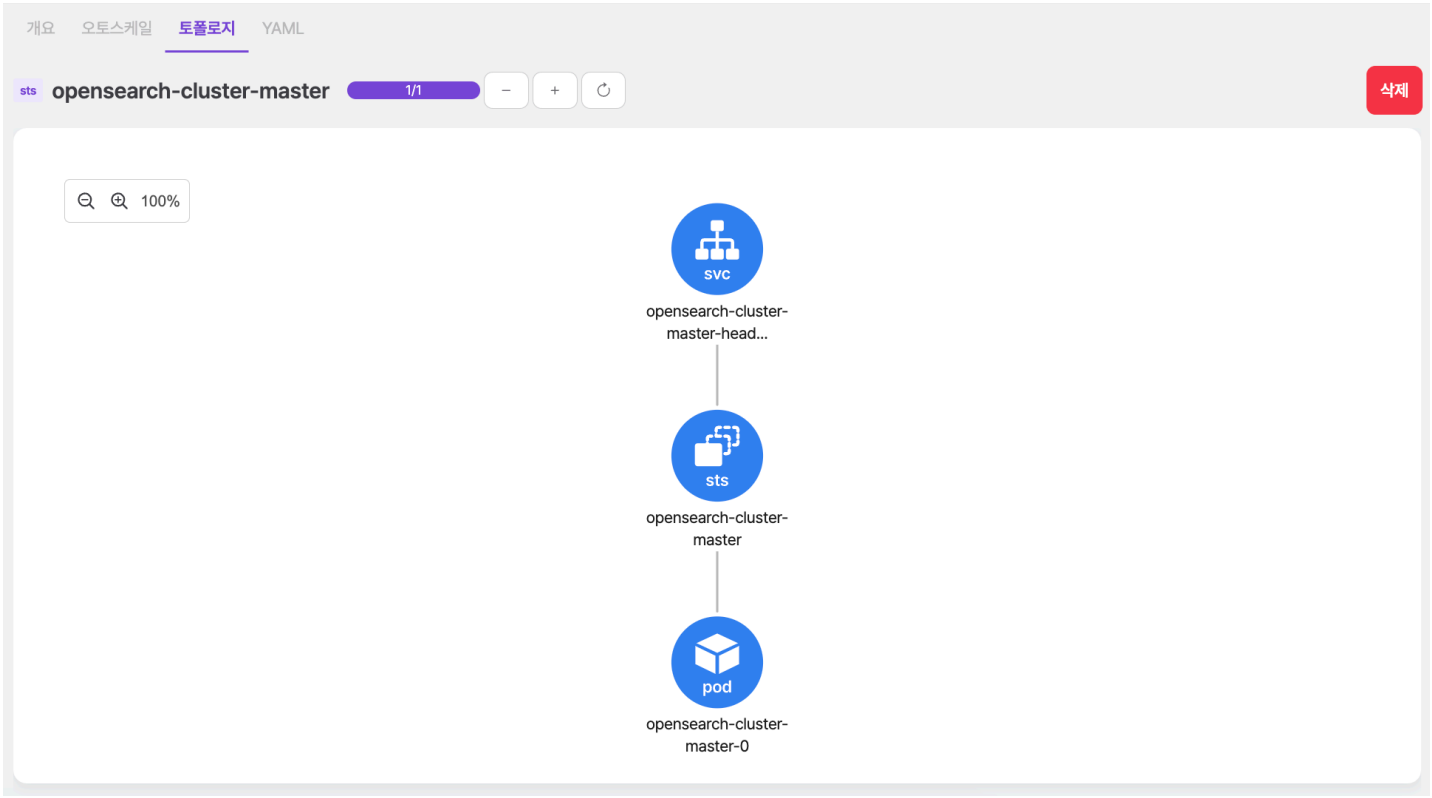
세부 정보
스테이트풀셋에 대한 세부 정보를 제공한다.




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

스테이트풀셋의 토폴로지에서 출력되는 리소스는 다음과 같다.

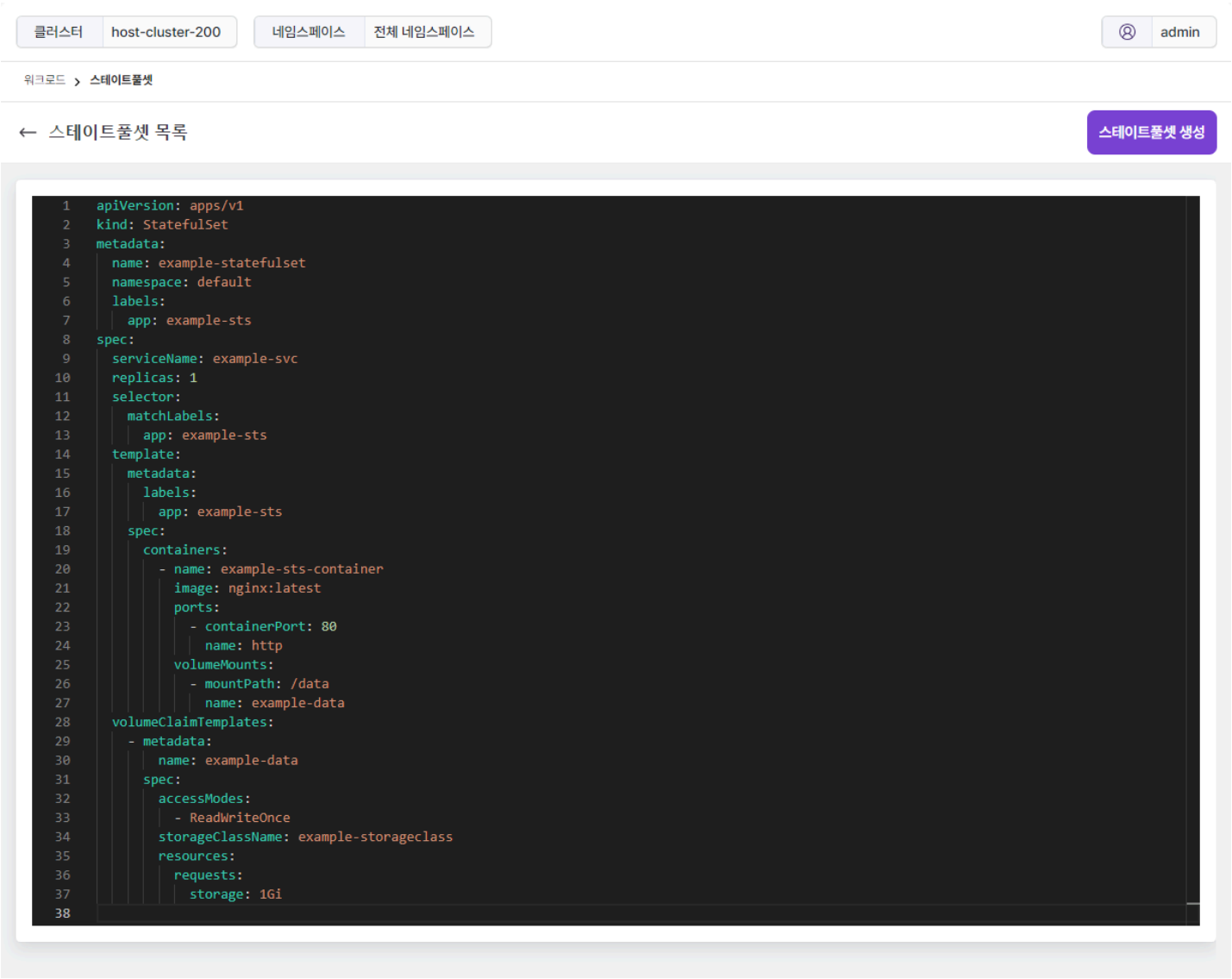
- 스테이트풀셋, 파드
- 서비스




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

## 스테이트풀셋 생성

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




## 스테이트풀셋 수정

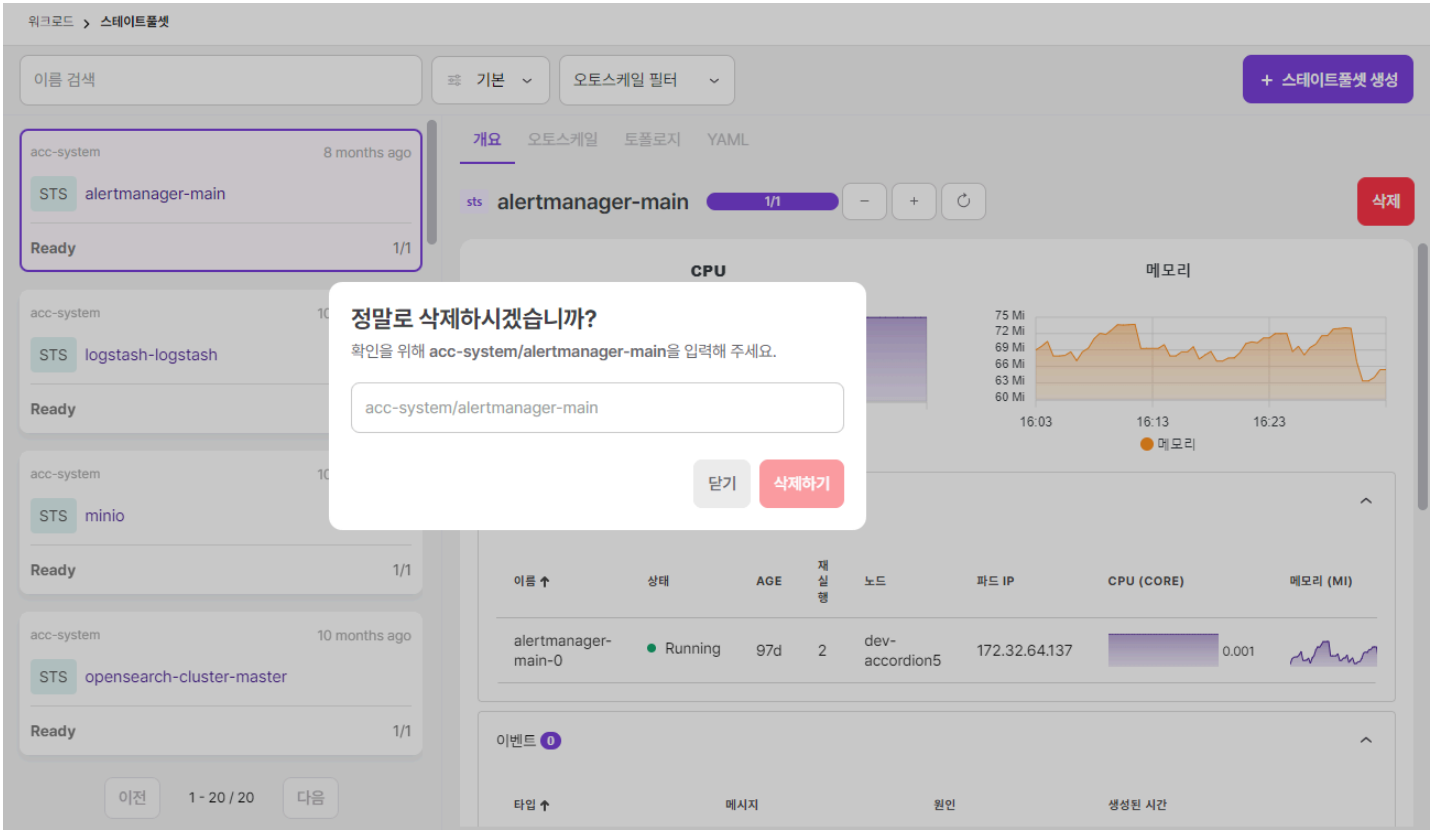
수정하려는 스테이트풀셋을 선택하고 우측의 YAML 편집기에서 정보를 변경 후 수정 버튼을 선택하여 반영한다.

스테이트풀셋도 디플로이먼트와 동일하게 오토스케일 설정이 가능하다.

> **TIP** 디플로이먼트 오토스케일 가이드 참고

## 스테이트풀셋 삭제

삭제하려는 스테이트풀셋을 선택하고 우측의 삭제 버튼을 선택한다.



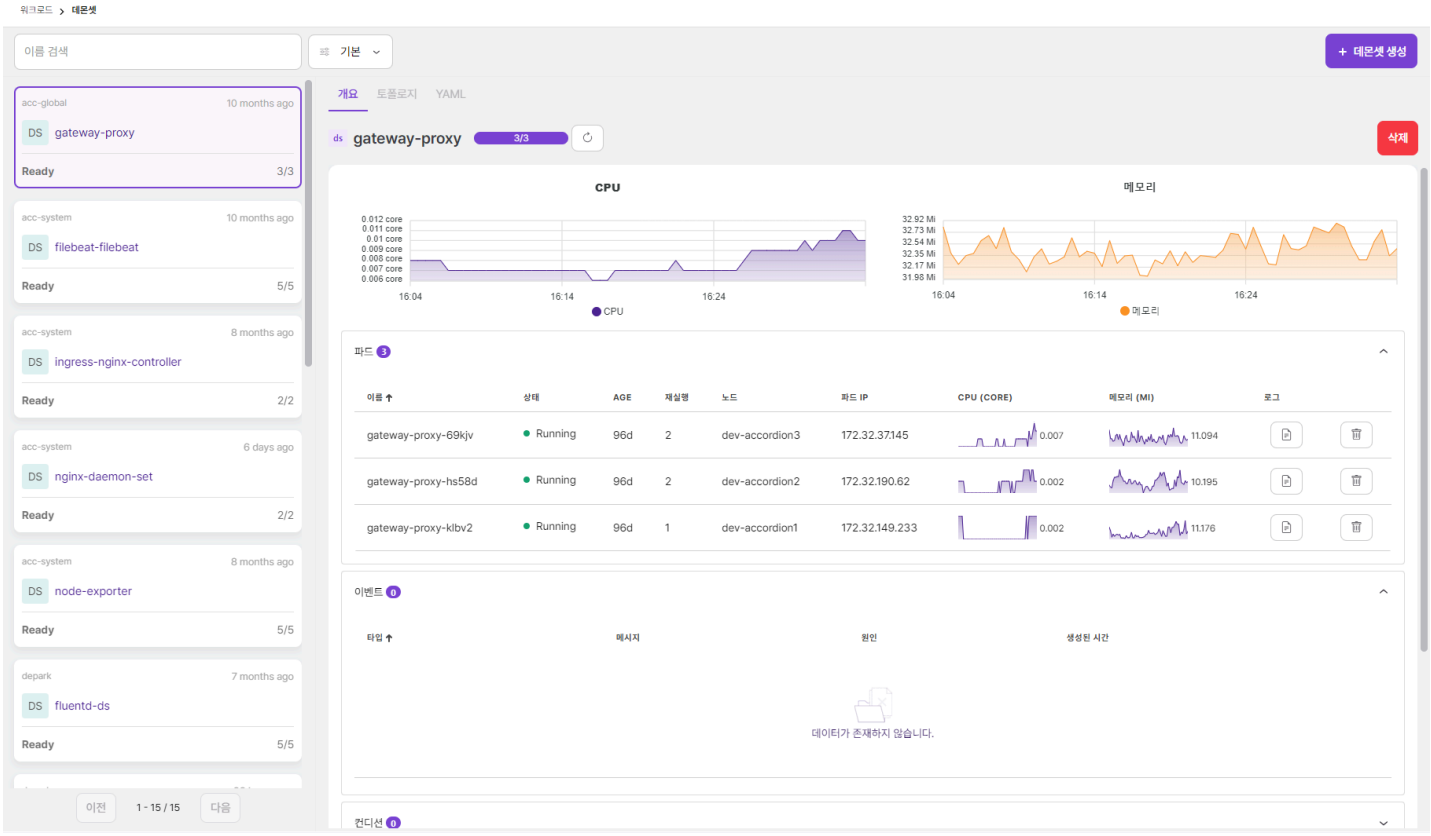

모달에서 네임스페이스와 스테이트풀셋 이름을 입력하여 삭제한다.

## 4.2.6.5. 데몬셋

배포된 데몬셋 정보를 제공한다.

### 개요

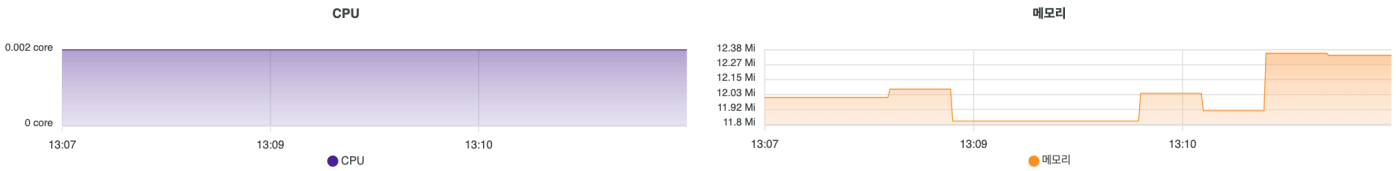
배포된 데몬셋의 메트릭, 파드, 이벤트, 상태, 세부 정보를 제공한다.




롤아웃

| 항목 | 설명 |
| --- | --- |
| 롤아웃 | 데몬셋을 롤링 재시작한다. |

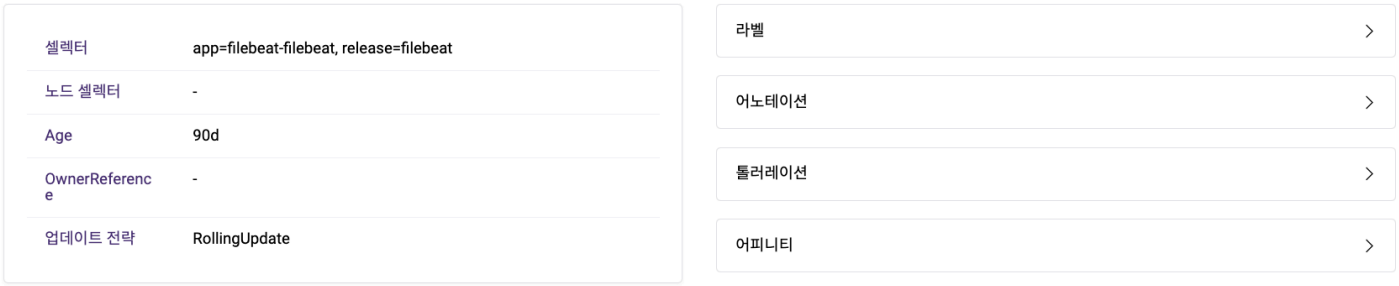
메트릭 정보
메트릭 정보는 현재 시간으로부터 5분간 CPU, Memory 사용량을 제공한다.



파드 정보
디플로이먼트와 내용이 동일하기 때문에 디플로이먼트 파드 정보를 참고한다.

이벤트 정보
파드와 내용이 동일하기 때문에 파드 이벤트 정보를 참고한다.

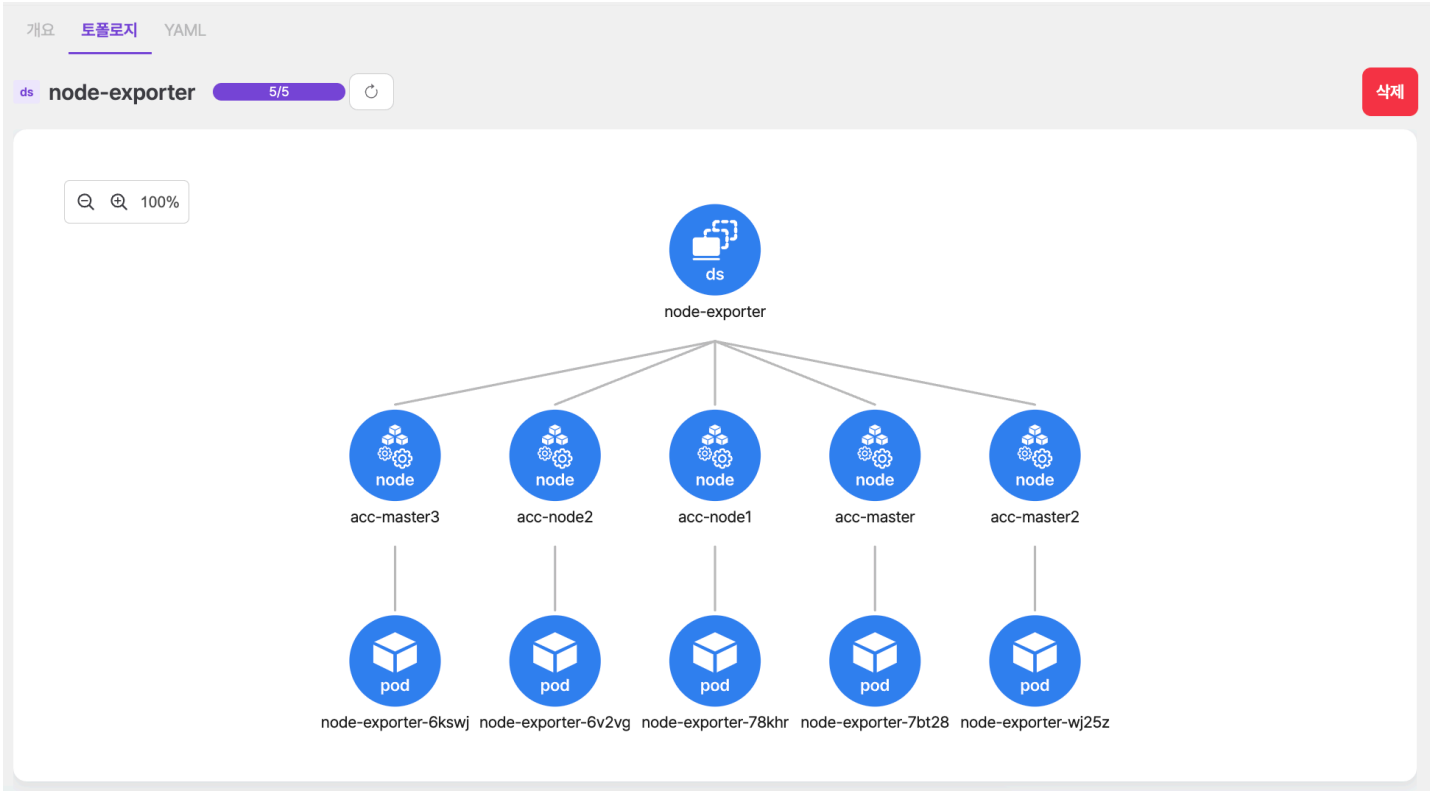
세부 정보
데몬셋에 대한 세부 정보를 제공한다.




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

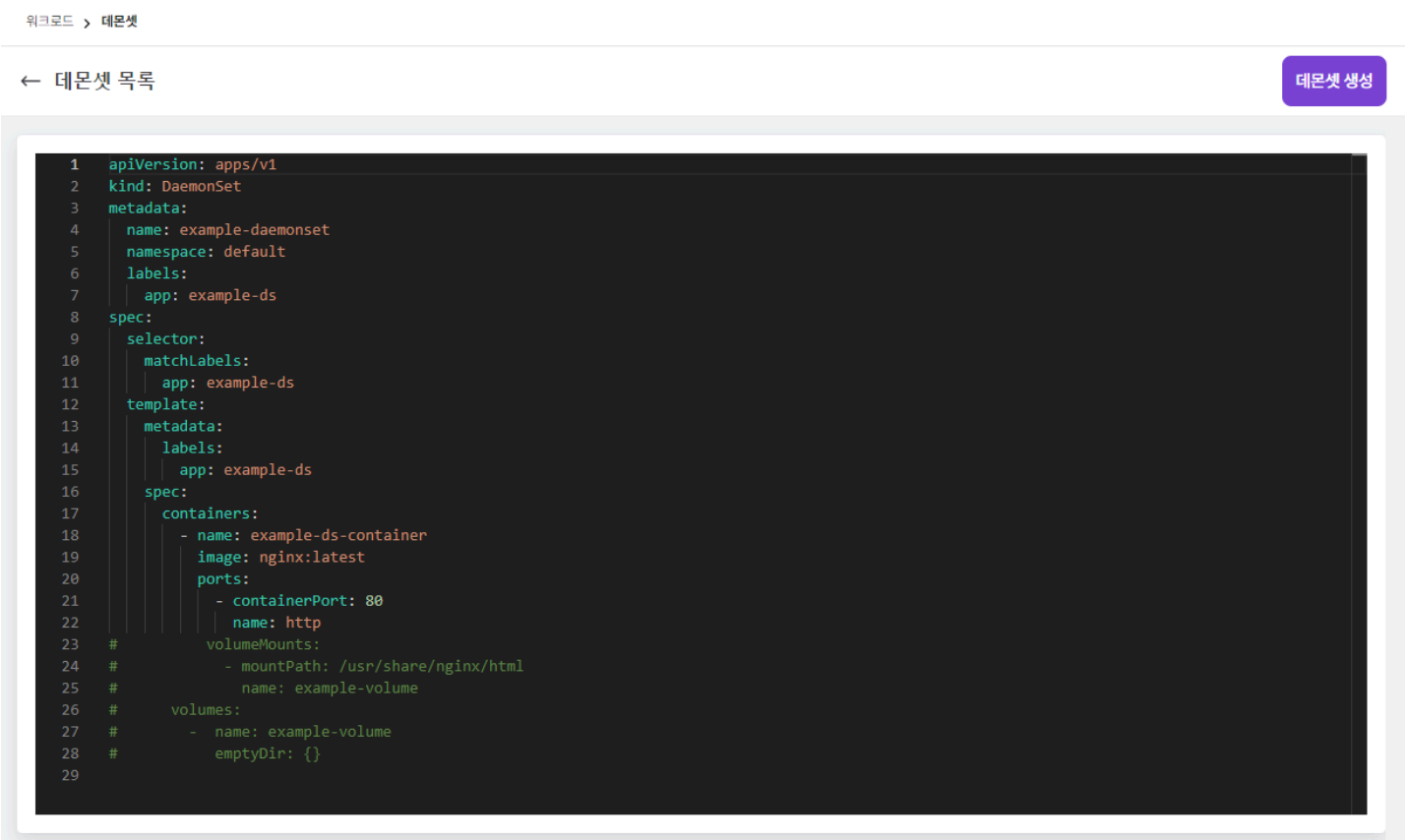
데몬셋의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 데몬셋, 노드, 파드



## 데몬셋 생성

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

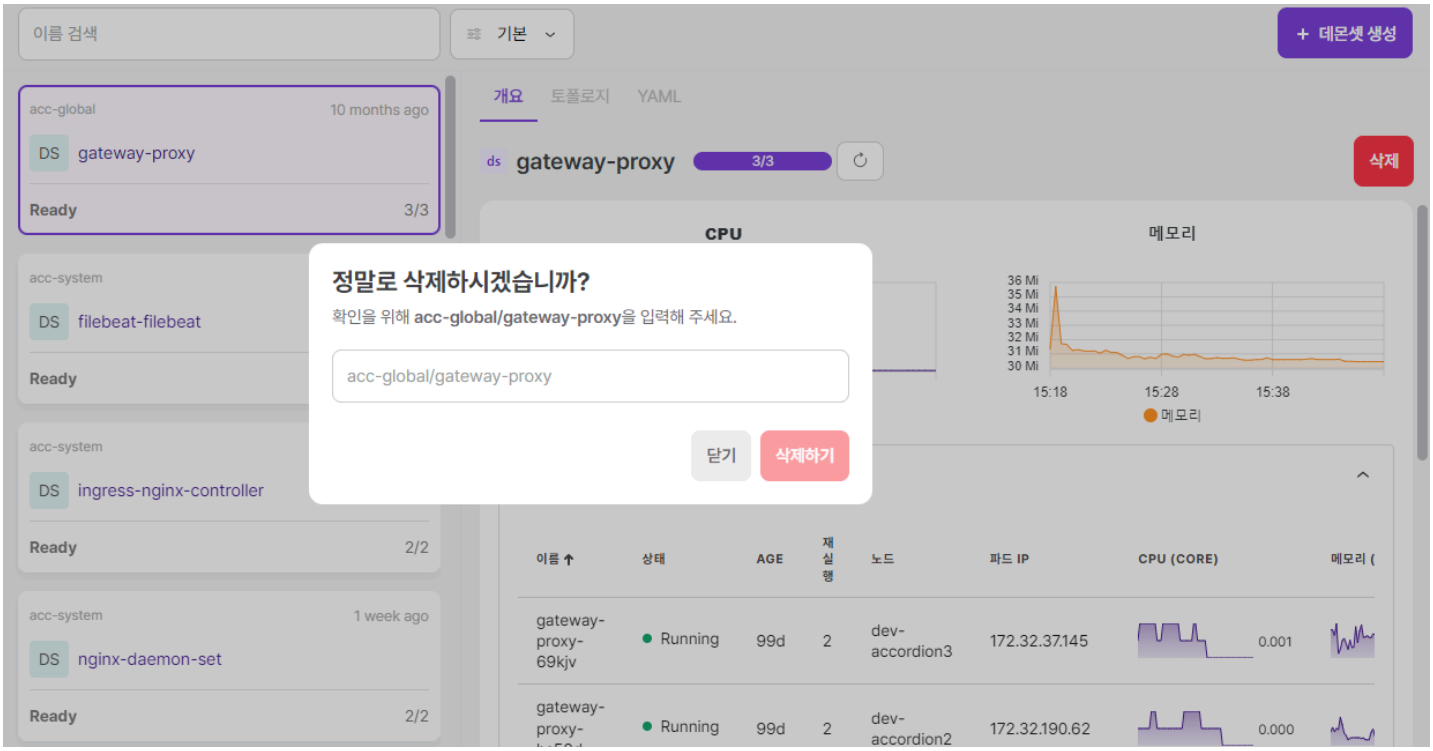



## 데몬셋 수정

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

## 데몬셋 삭제

삭제하려는 데몬셋을 선택하고 우측의 삭제 버튼을 선택한다.



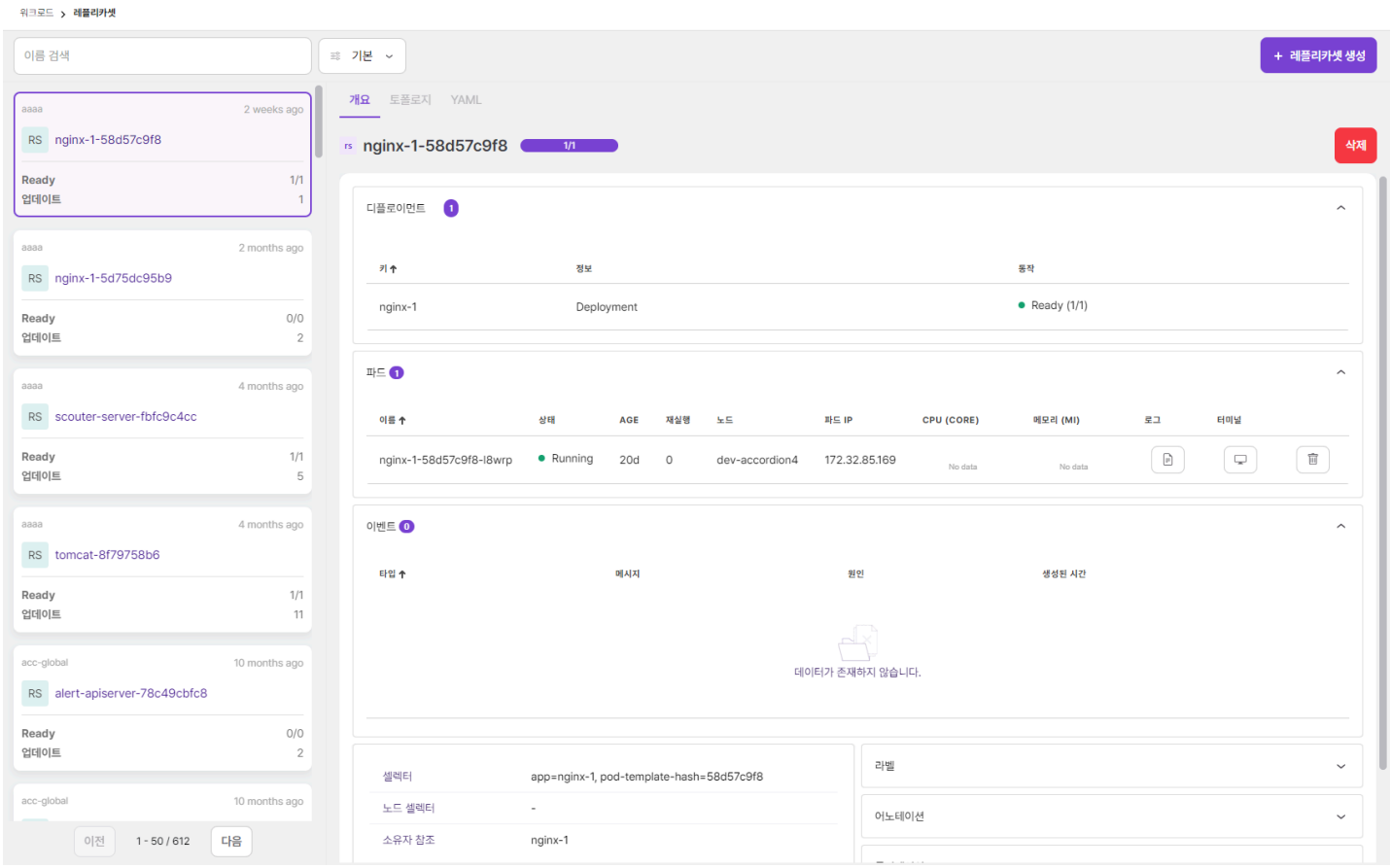
모달에서 네임스페이스와 데몬셋 이름을 입력하여 삭제한다.

## 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.

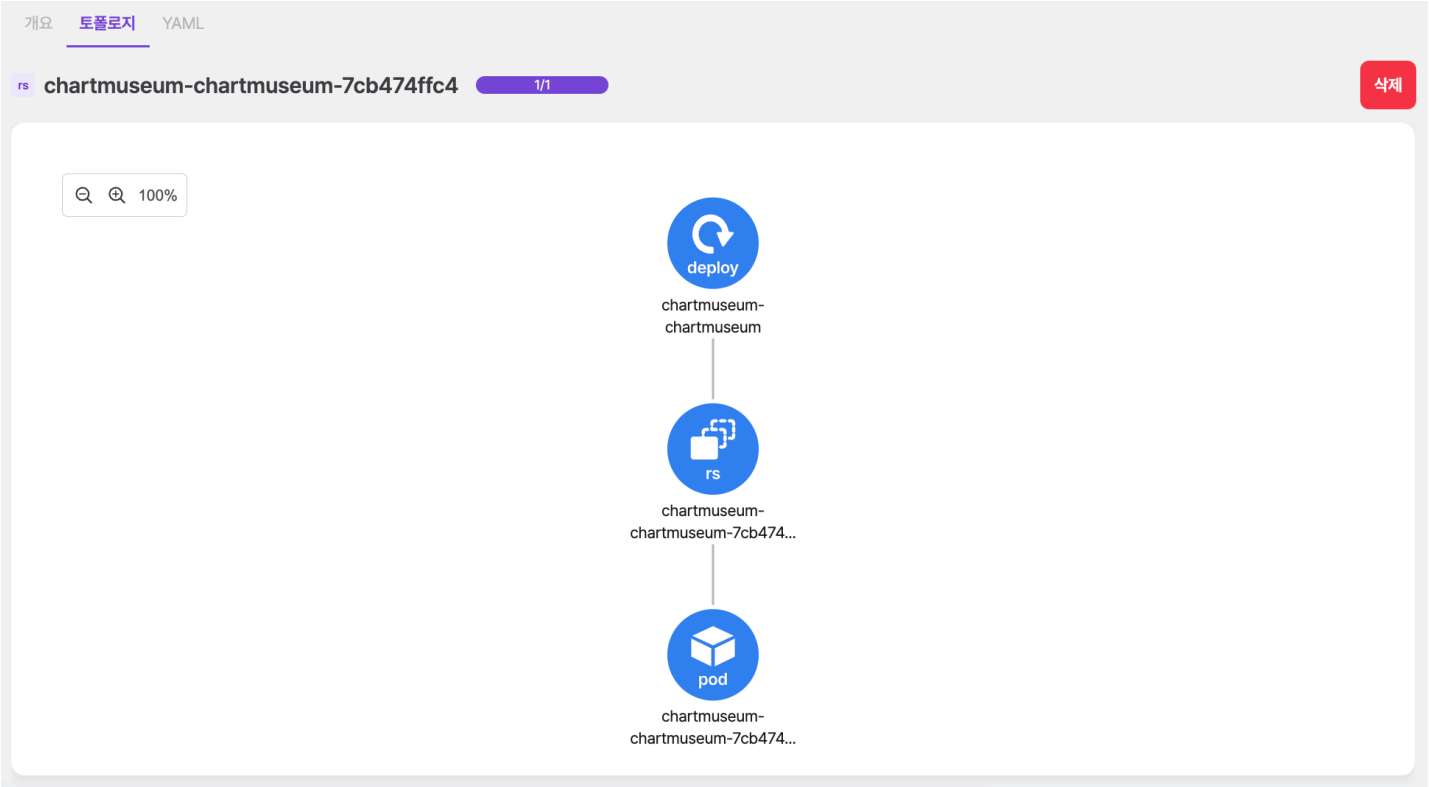
이벤트 정보
파드와 내용이 동일하기 때문에 파드 이벤트 정보를 참고한다.

세부 정보
레플리카셋에 대한 세부 정보를 제공한다.

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

레플리카셋의 토폴로지에서 출력되는 리소스는 다음과 같다.

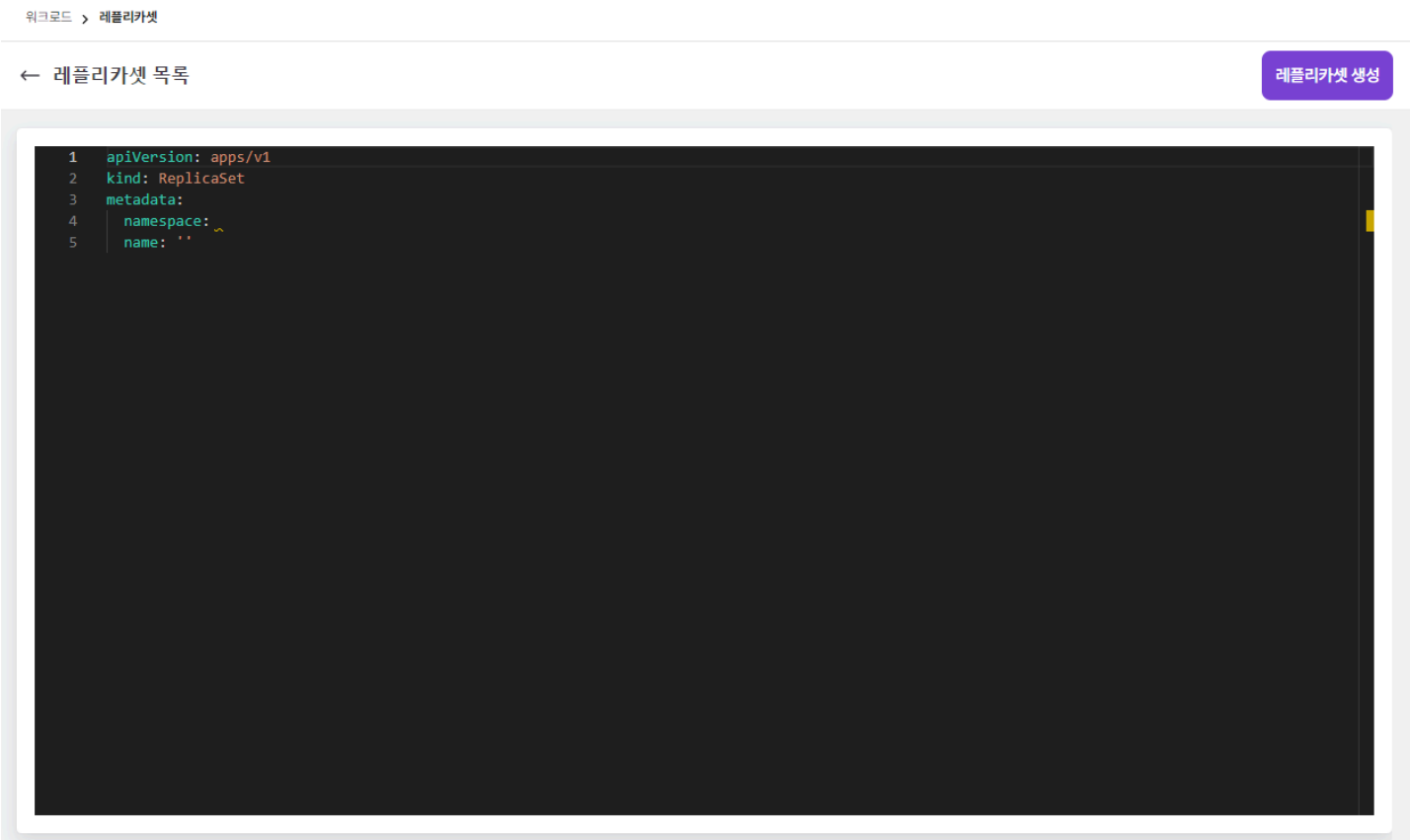
- 디플로이먼트, 레플리카셋, 파드




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.

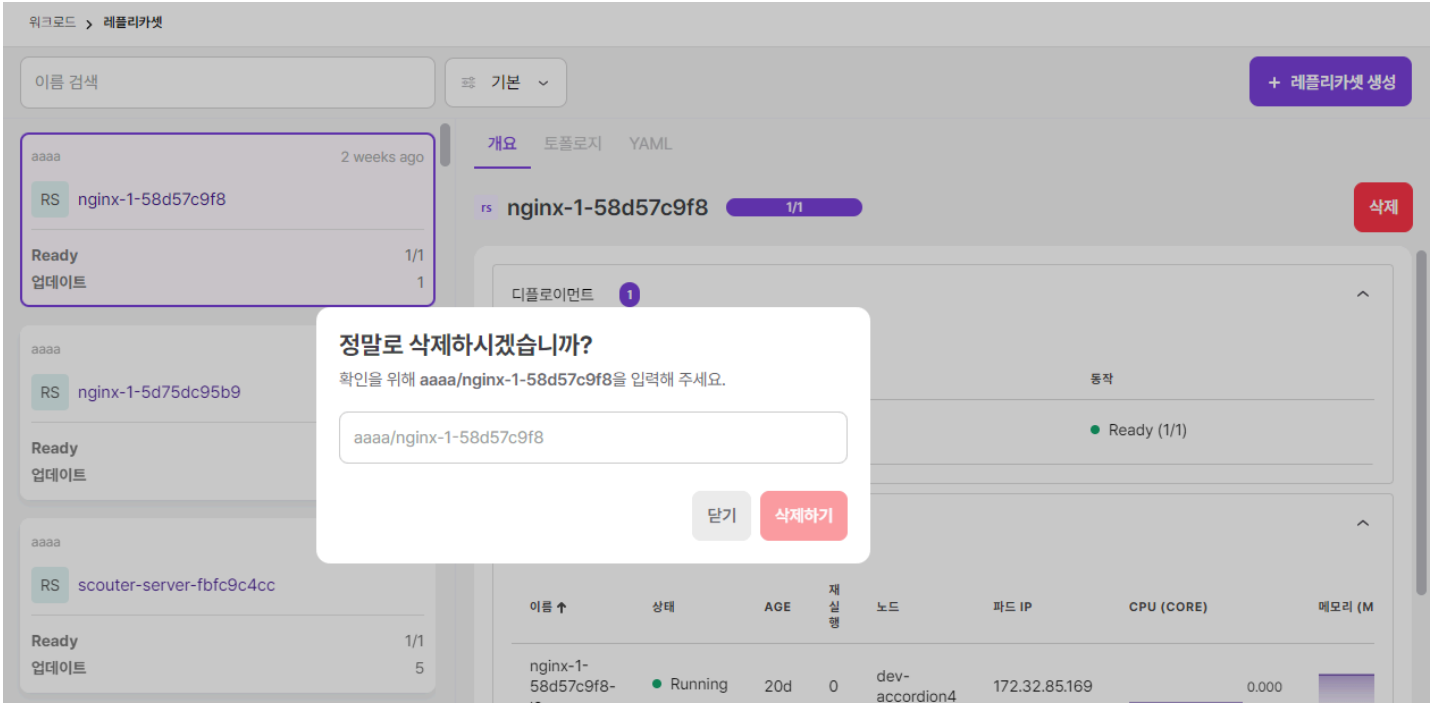



## 레플리카셋 수정

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

## 레플리카셋 삭제

삭제하려는 레플리카셋을 선택하고 우측의 삭제 버튼을 선택한다.



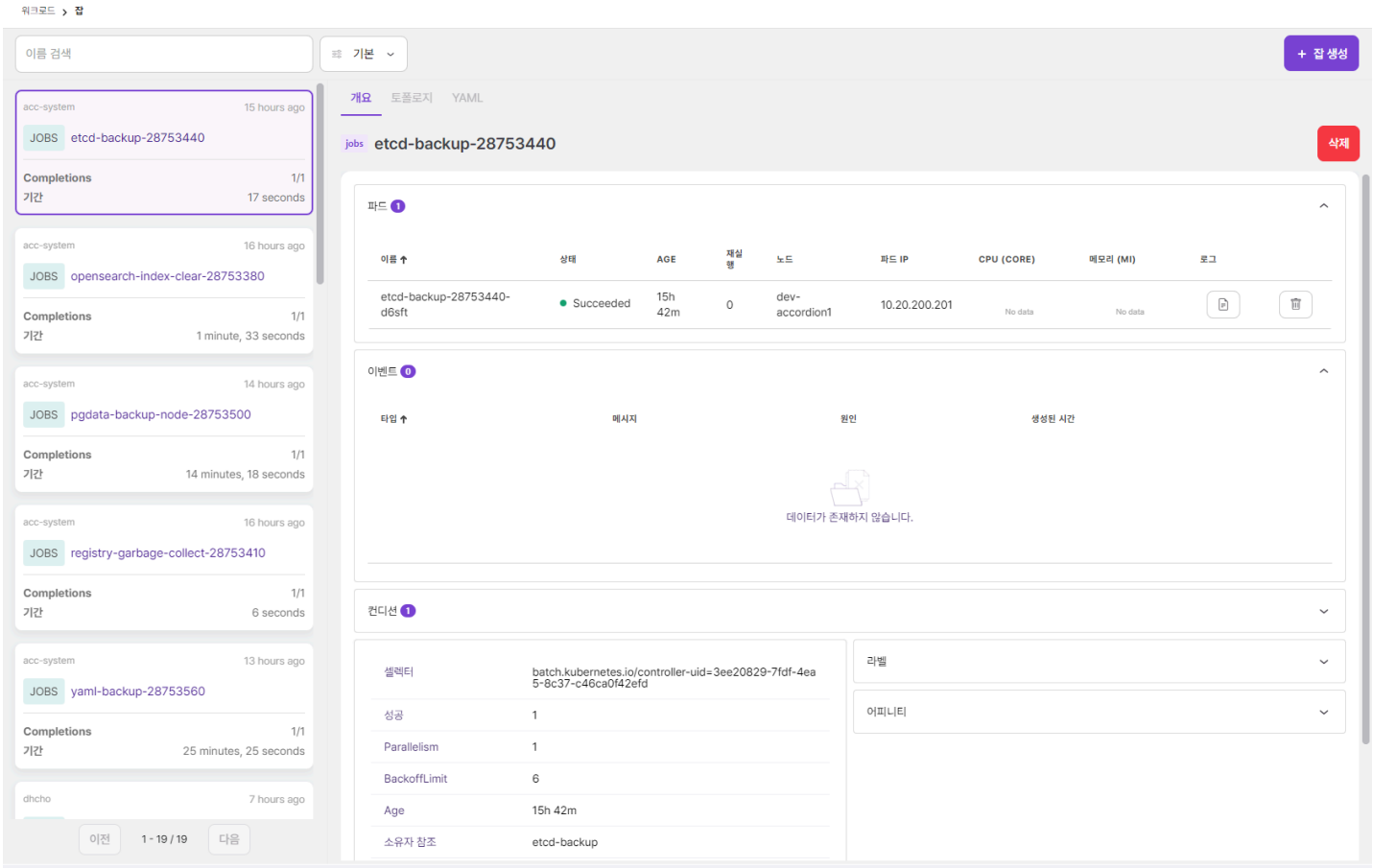

모달에서 네임스페이스와 레플리카셋 이름을 입력하여 삭제한다.

## 4.2.6.7. 잡

배포된 잡 정보를 제공한다.

### 개요

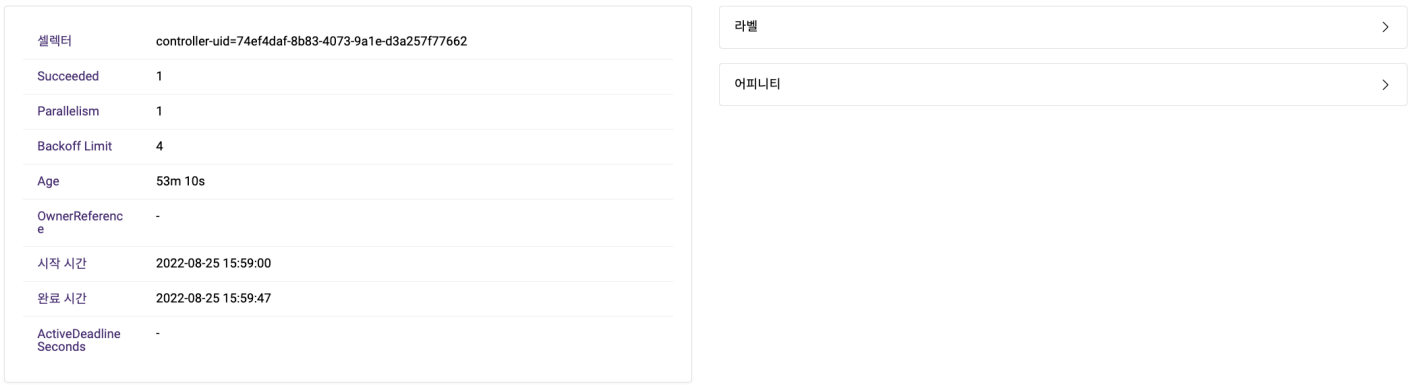
배포된 잡의 파드, 이벤트, 상태, 세부 정보를 제공한다.




파드 정보
디플로이먼트와 내용이 동일하기 때문에 디플로이먼트 파드 정보를 참고한다.

이벤트 정보
파드와 내용이 동일하기 때문에 파드 이벤트 정보를 참고한다.

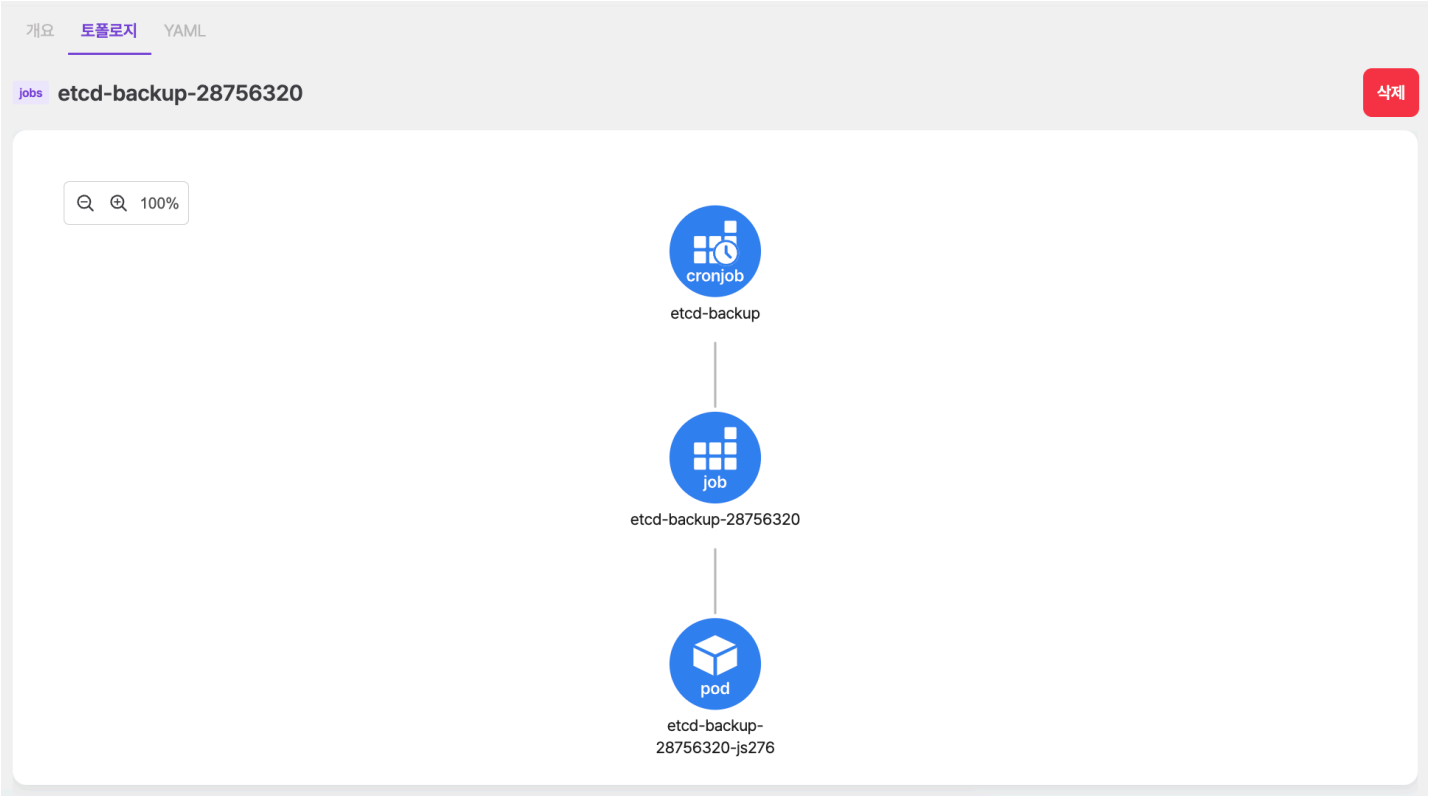
세부 정보
잡에 대한 세부 정보를 제공한다.




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

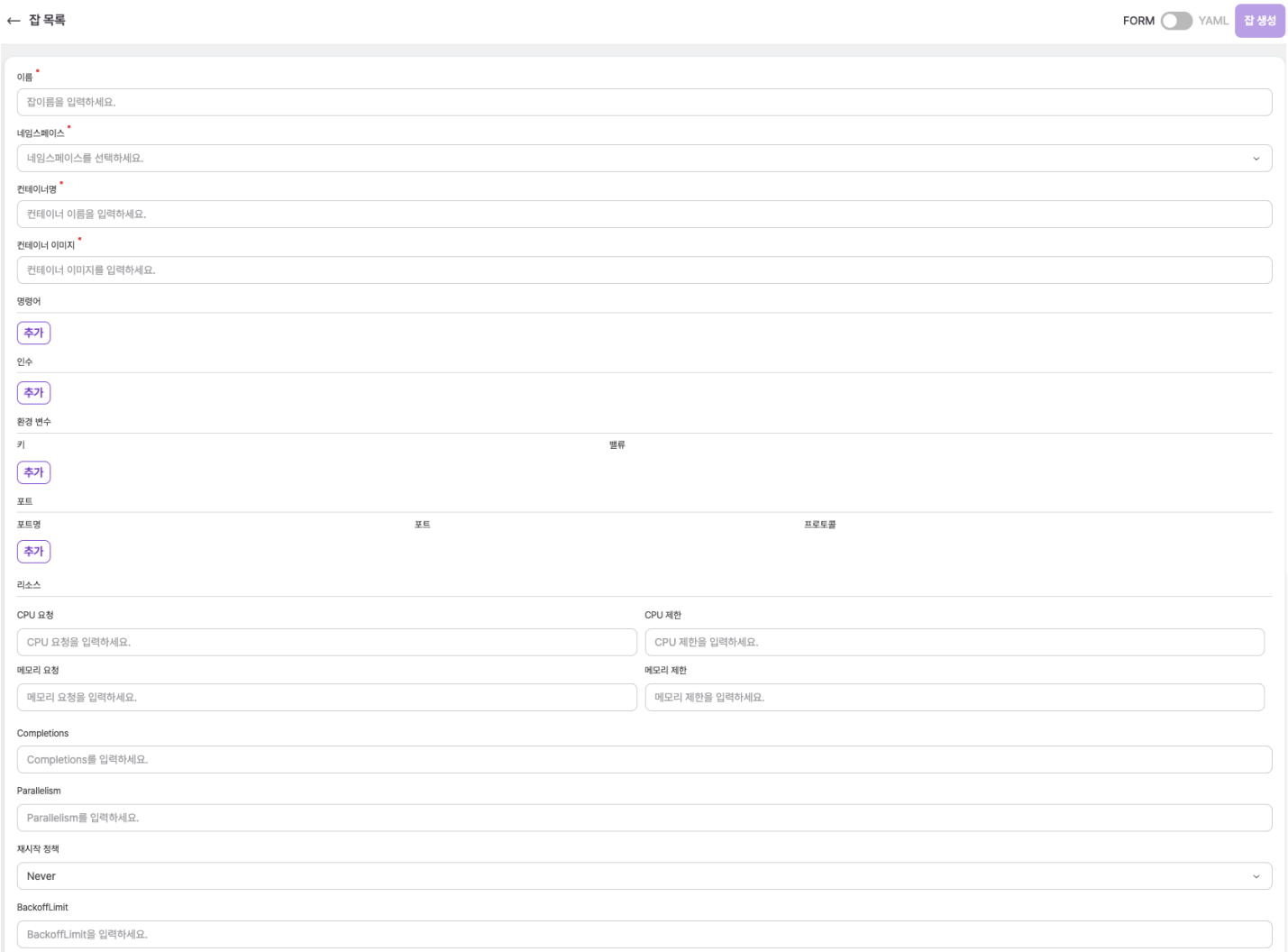
잡의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 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.




| 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 |

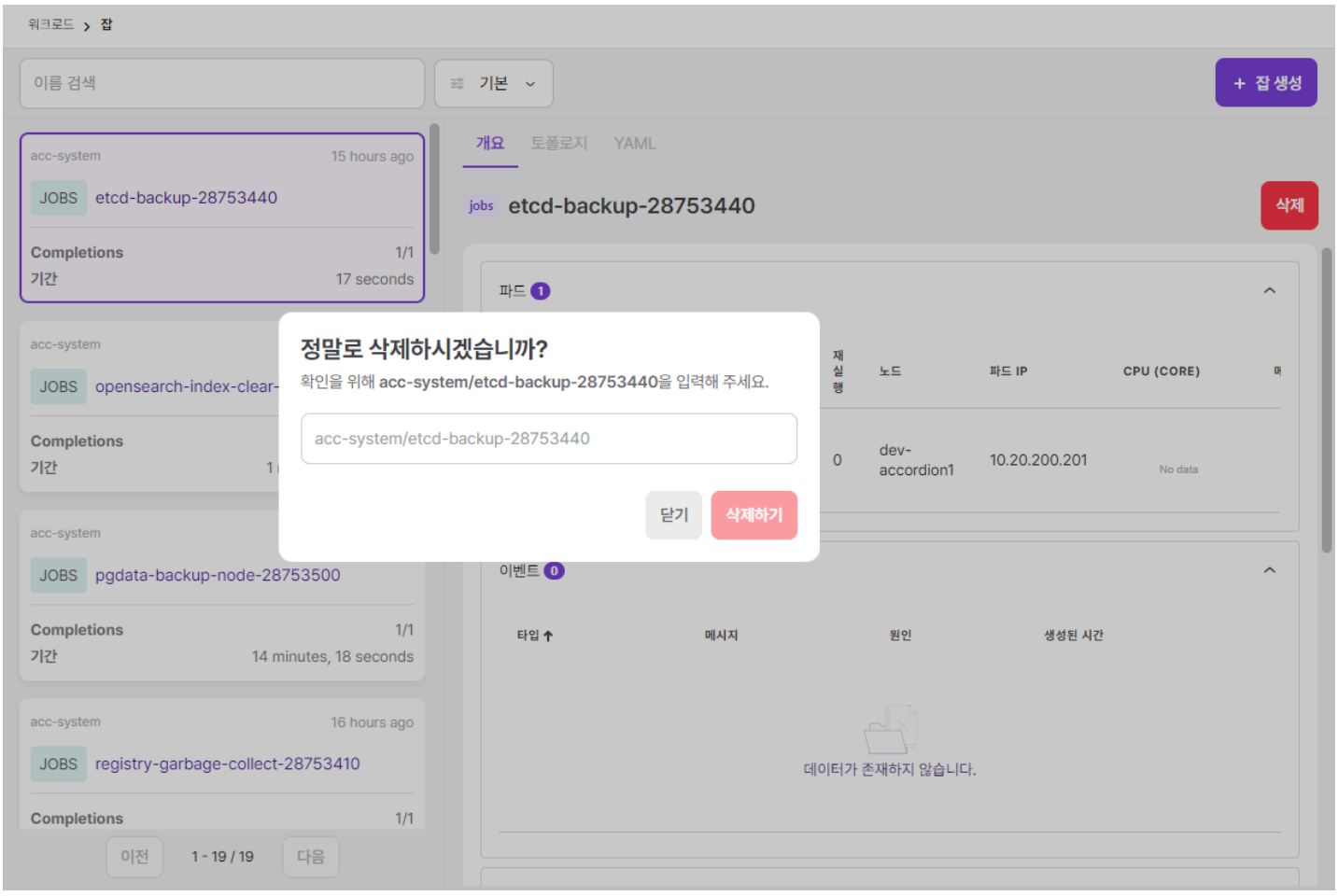
| 항목 | 설명 |
| --- | --- |
| 리소스 | 컨테이너에 할당될 cpu,memory 리소스<br>• 요청(Requests) : 최소 필요량<br>• 제한(Limits) : 최대 허용량을 지정 |
| 완료 | 잡이 성공적으로 완료되어야 하는 파드 개수 |
| 병렬 처리 | 동시에 실행될 파드 인스턴스의 수 |
| 재시작 정책 | 잡이 실패할 경우 재시작 여부와 정책 설정<br>• OnFailure : 실패 시 재시작<br>• Never : 실패 시 재시작하지 않음 |
| 백오프 한계 | 잡의 파드의 재시도 횟수 제한, 초과시 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.



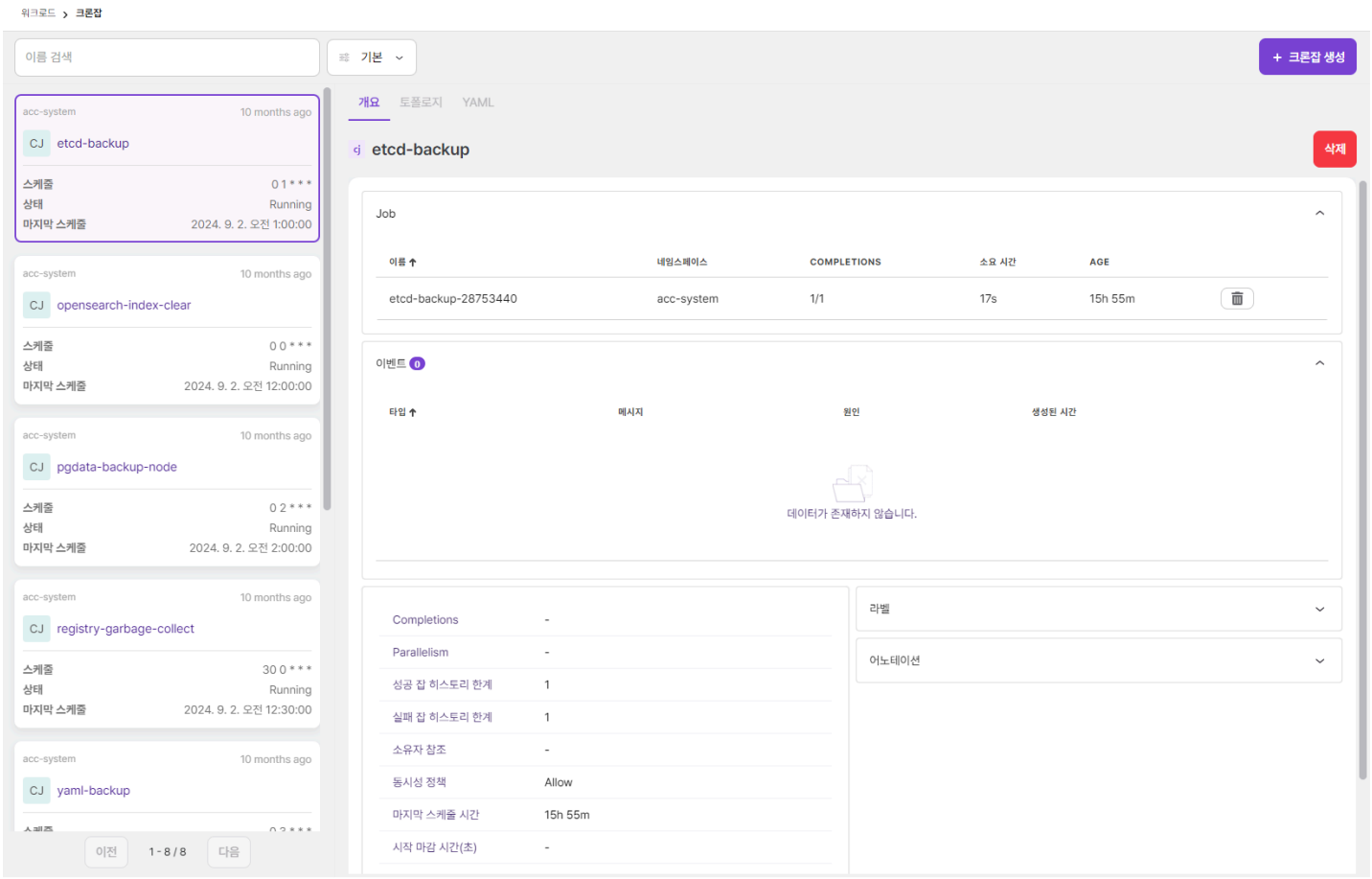

모달에서 네임스페이스와 잡 이름을 입력하여 삭제한다.

## 4.2.6.8. 크론잡

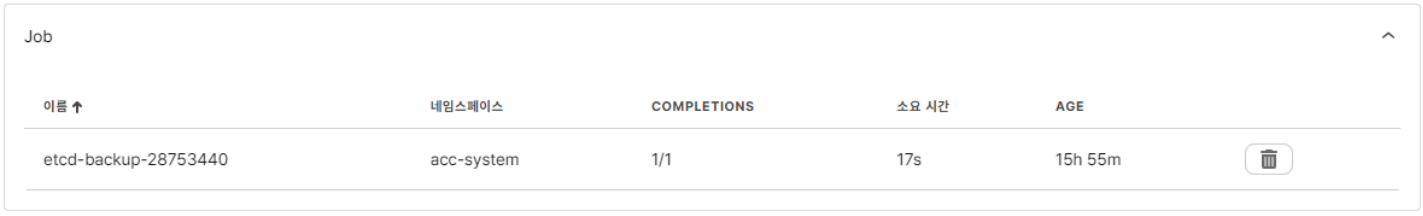
배포된 크론잡 정보를 제공한다.

### 개요

배포된 크론잡의 잡, 이벤트, 세부정보를 제공한다.




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

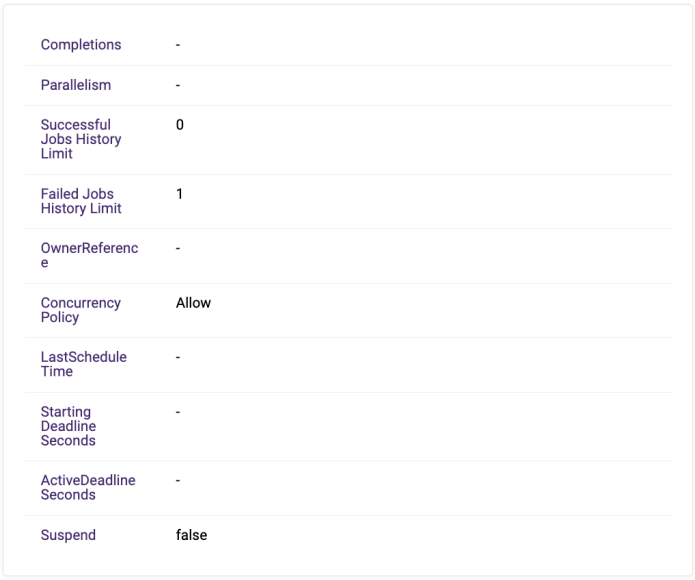



| 항목 | 설명 |
| --- | --- |
| 이름 | 잡 이름 |
| 네임스페이스 | 잡이 생성된 네임스페이스 |
| COMPLETIONS | 크론잡으로 생성된 잡 완료 개수 |
| 소요 시간 | 잡이 실행되고 완료되기까지 시간 |
| AGE | 잡이 생성되고나서 지난 시간 |

| 항목 | 설명 |
| --- | --- |
| 삭제 | 잡 삭제 |

이벤트 정보
파드와 내용이 동일하기 때문에 파드 이벤트 정보를 참고한다.

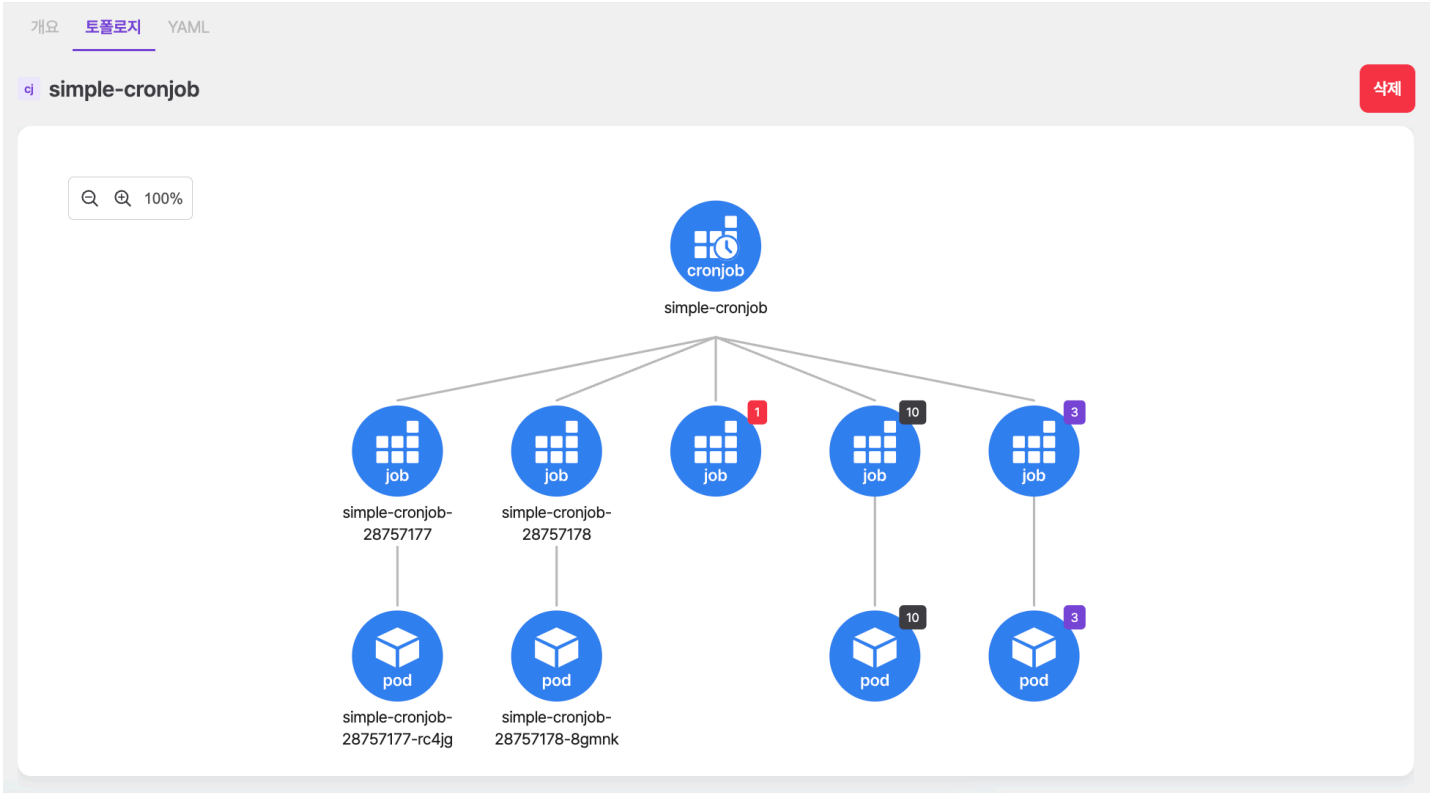
세부 정보
크론잡에 대한 세부 정보를 제공한다.




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

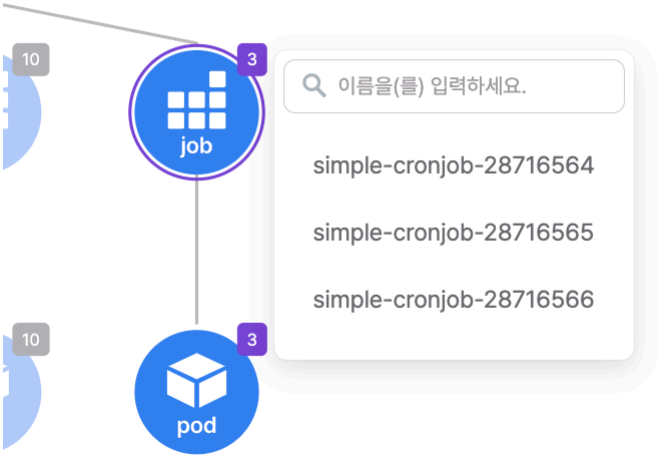

개요
토폴로지
YAML
simple-cronjob
삭제
100%
cronjob
simple-cronjob
job
simple-cronjob-28757177
job
simple-cronjob-28757178
job
1
job
10
job
3
pod
simple-cronjob-28757177-rc4jg
pod
simple-cronjob-28757178-8gmnk
pod
10
pod
3

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 |

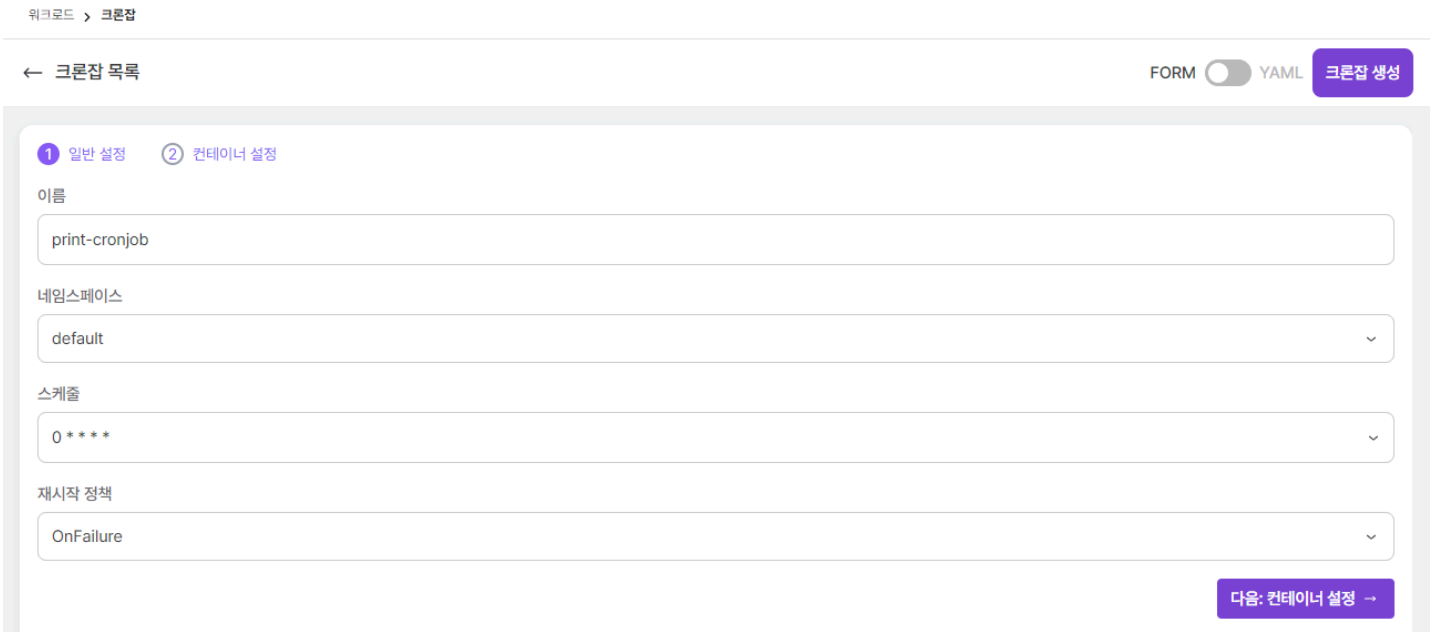
개수를 출력하는 노드를 클릭할 경우, 해당하는 상태의 잡 또는 파드 리스트를 드롭다운으로 출력하며 클릭할 경우 해당 리소스 화면으로 이동한다.



## 크론잡 생성

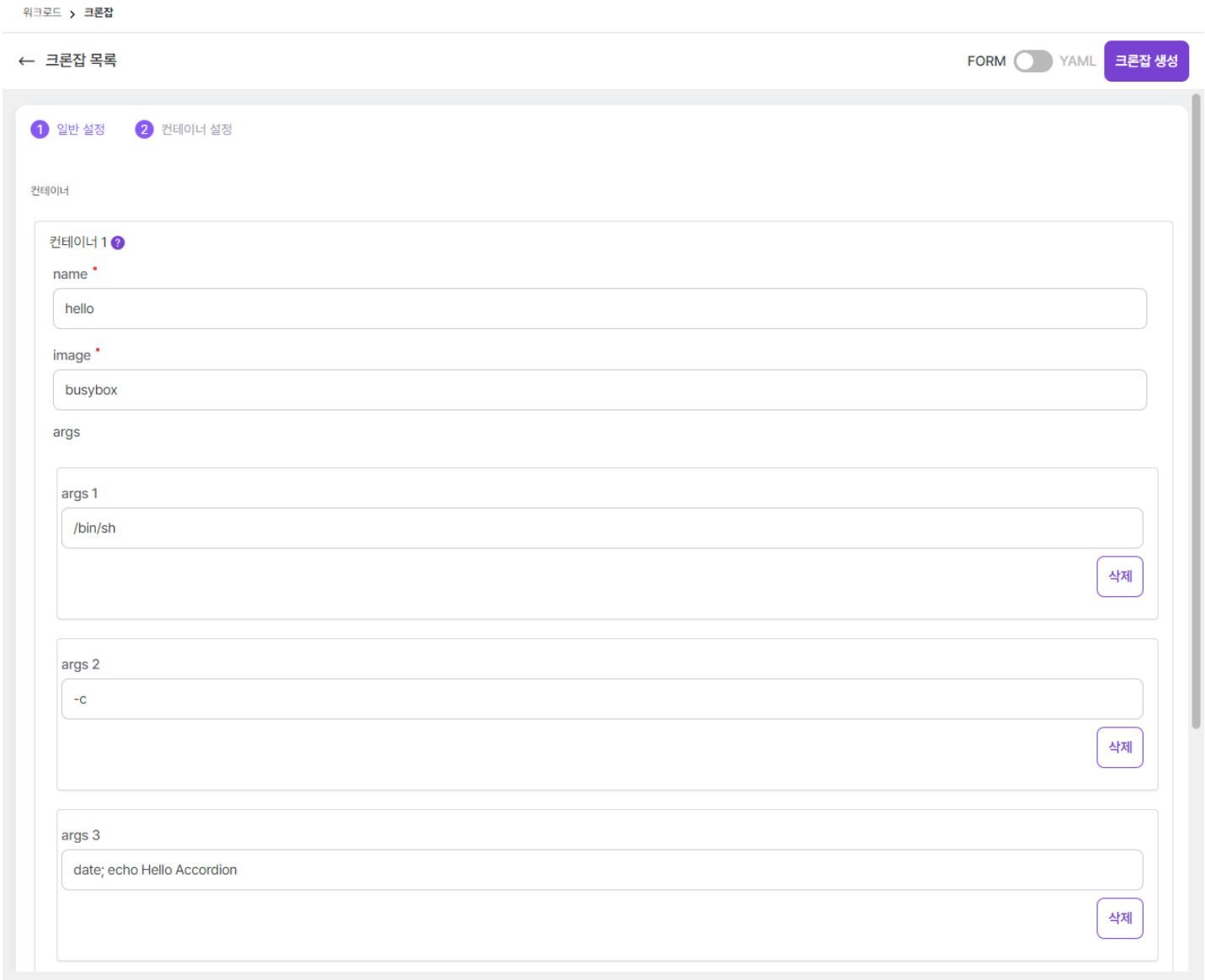
+ 크론잡 생성 을 선택하면 나타나는 화면에서 쿠버네티스 크론잡 리소스 정보를 입력하여 생성할 수 있다. 생성 시에는 FORM/YAML로 입력할 수 있으며 일반 설정과 컨테이너 설정을 하게 된다.

### 일반 설정




| 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<br>• OnFailure는 실패 시 재시작<br>• Never는 재시작하지 않음 |

컨테이너 설정



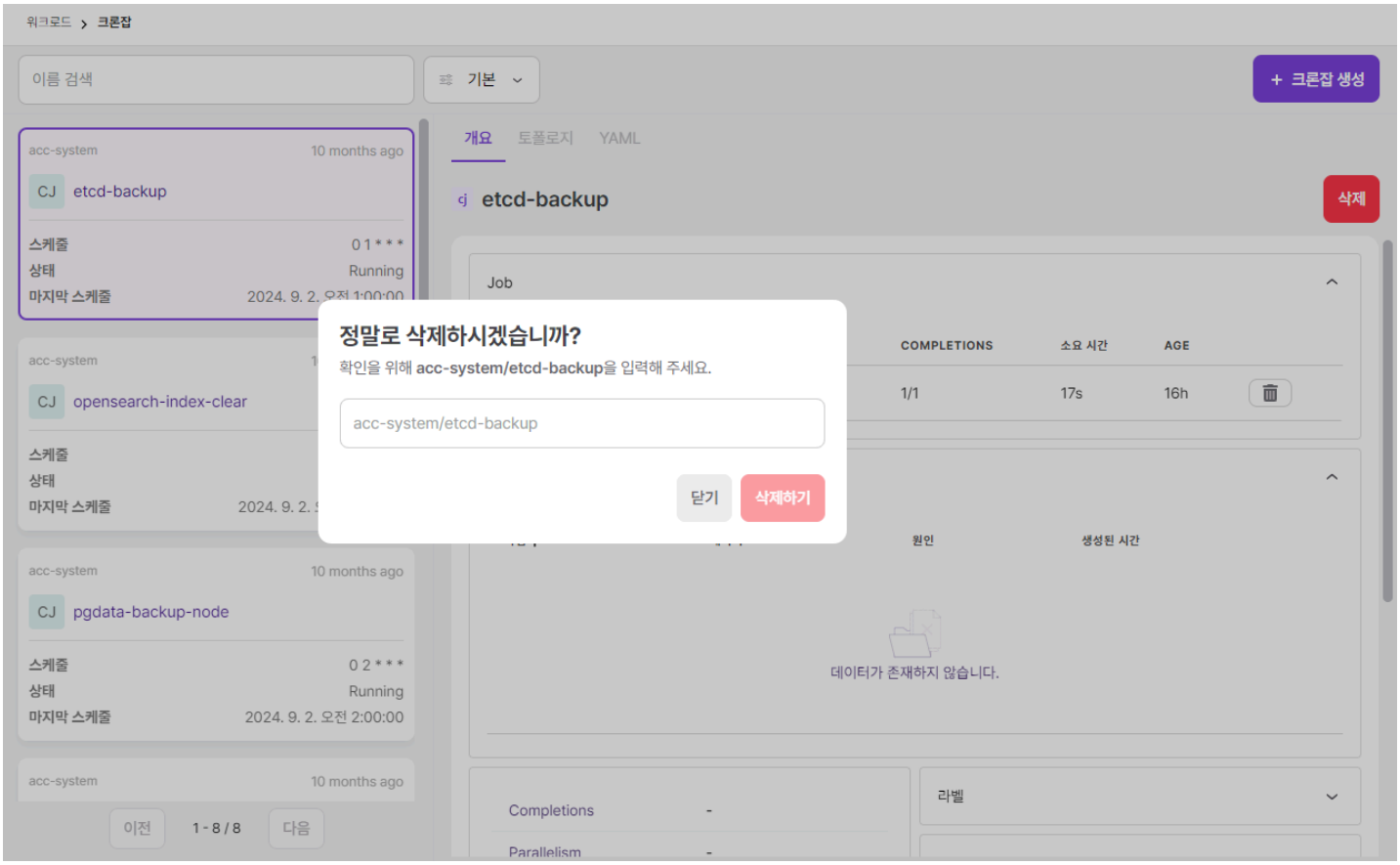

| 항목 | 설명 |
| --- | --- |
| name | 크론잡에서 실행될 컨테이너 이름 |
| image | 크론잡에서 실행될 컨테이너 이미지 |
| args | 컨테이너가 시작될 때 실행될 명령어 |

## 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

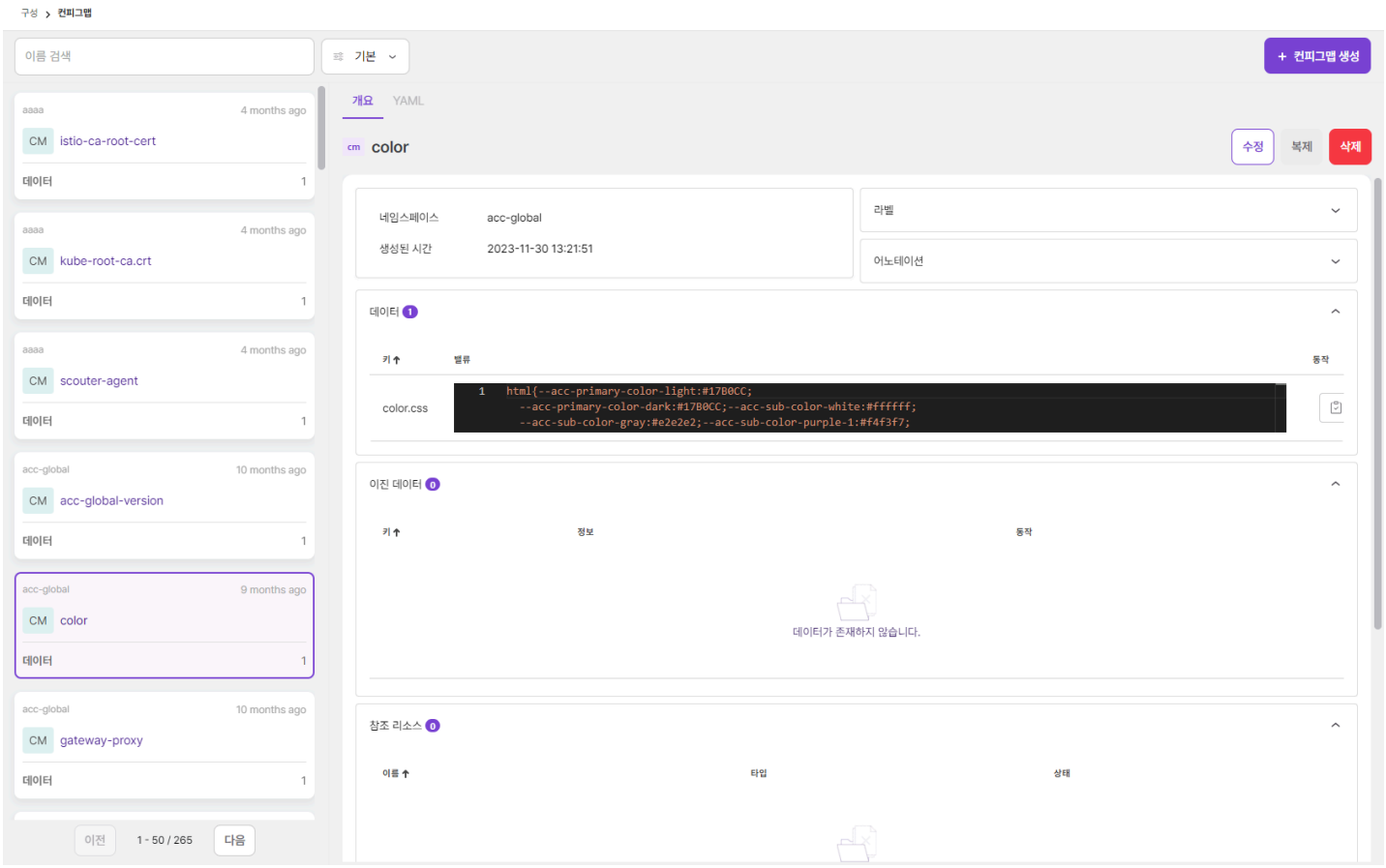
컨피그맵과 시크릿은 애플리케이션(또는 워크로드)가 정상적으로 실행되기 위해 필요한 구성요소 정보를 저장한다. 그 외에도 오토스케일 설정을 위한 HPA와 리소스 제한을 위한 리밋레인지 설정을 할 수 있다.

### 4.2.7.1. 컨피그맵

컨피그맵은 키-값 쌍으로 기밀이 아닌 데이터를 저장한다.

#### 개요

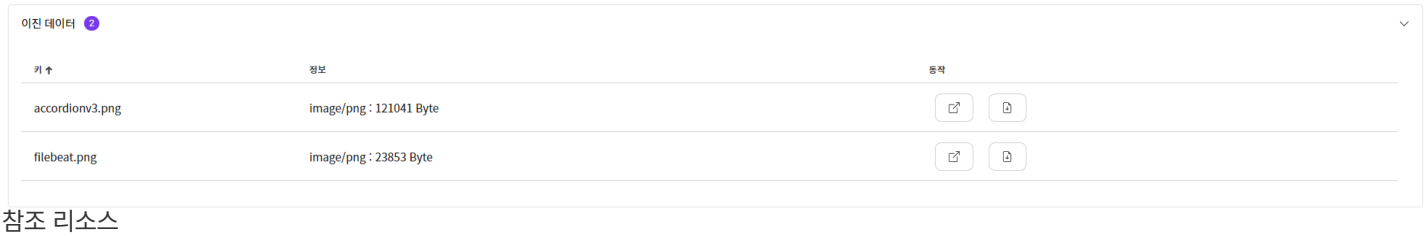
배포된 컨피그맵의 세부 정보, 데이터, 이진 데이터, 참조 리소스, 이벤트를 제공한다.



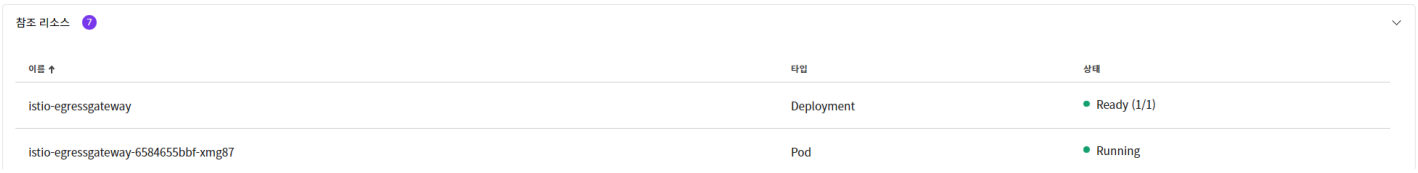

세부 정보
컨피그맵에 대한 세부 정보를 제공한다.

데이터
컨피그맵의 데이터 정보를 제공하며 우측에 복사 버튼을 제공한다.

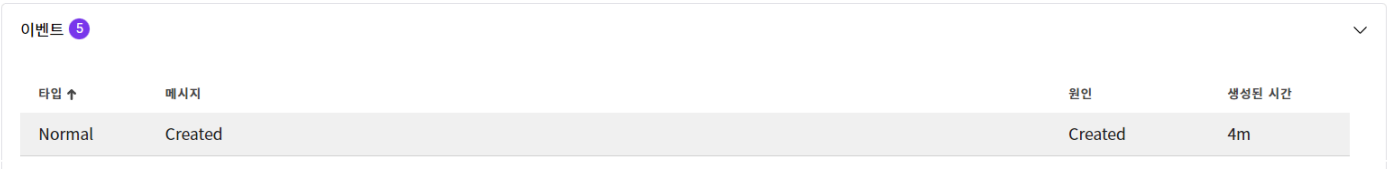
이진 데이터
컨피그맵의 이진 데이터 정보를 제공하며 우측에 미리보기와 다운로드 버튼을 제공한다.




참조 리소스
컨피그맵의 참조 리소스 정보를 제공하며 클릭 시 해당 리소스로 이동된다.




이벤트 정보
컨피그맵에서 발생하는 이벤트 정보를 제공한다.




| 항목 | 설명 |
| --- | --- |
| 타입 | 이벤트 타입<br>• Normal: 일반적인 작업으로 발생된 이벤트<br>• Warning: 오류에 의해 발생된 이벤트 |
| 메시지 | 이벤트 메시지 |
| 원인 | 이벤트 발생 이유 |
| 생성된 시간 | 이벤트 생성 후 지난 시간 |

## 컨피그맵 생성

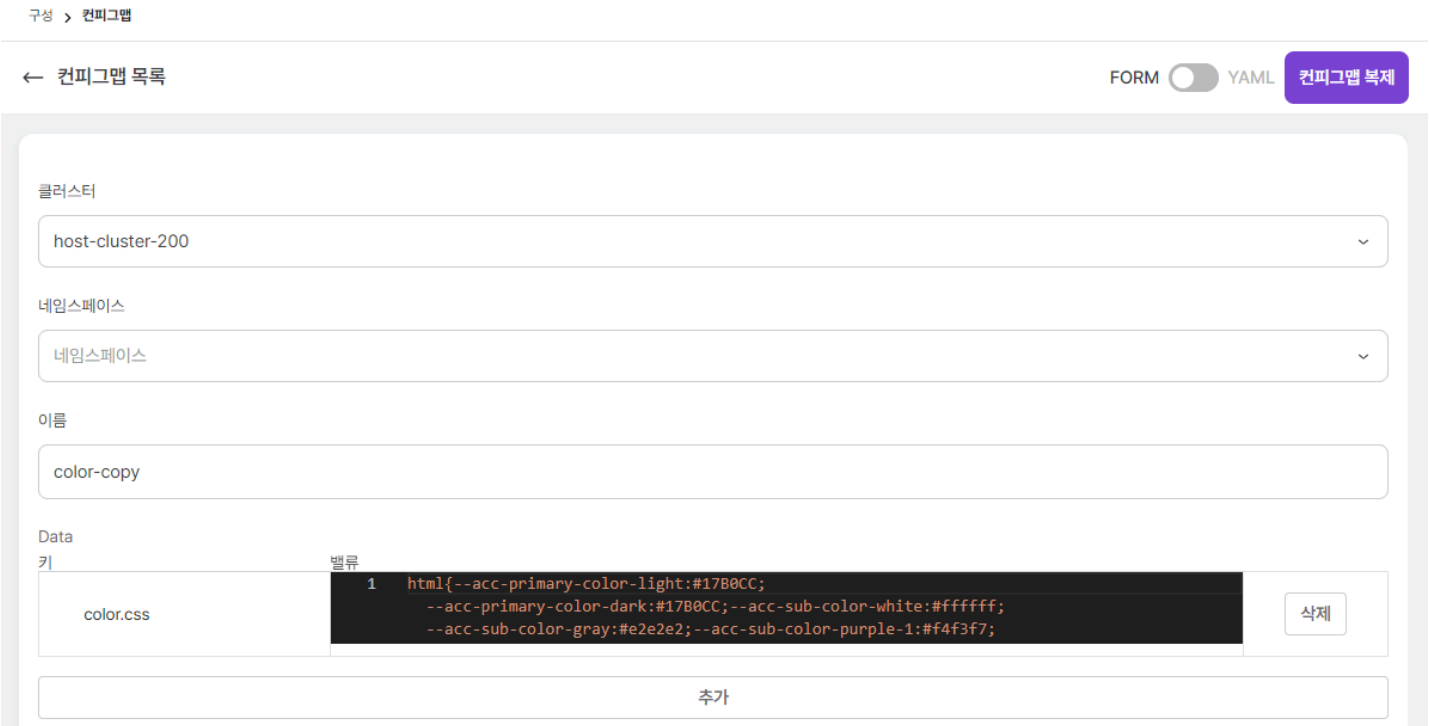
+ 컨피그맵 생성 을 선택하면 나타나는 화면에서 쿠버네티스 컨피그맵 리소스 정보를 입력하여 생성할 수 있다. 생성 시에는 FORM/YAML로 입력할 수 있다.

## 컨피그맵 수정

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

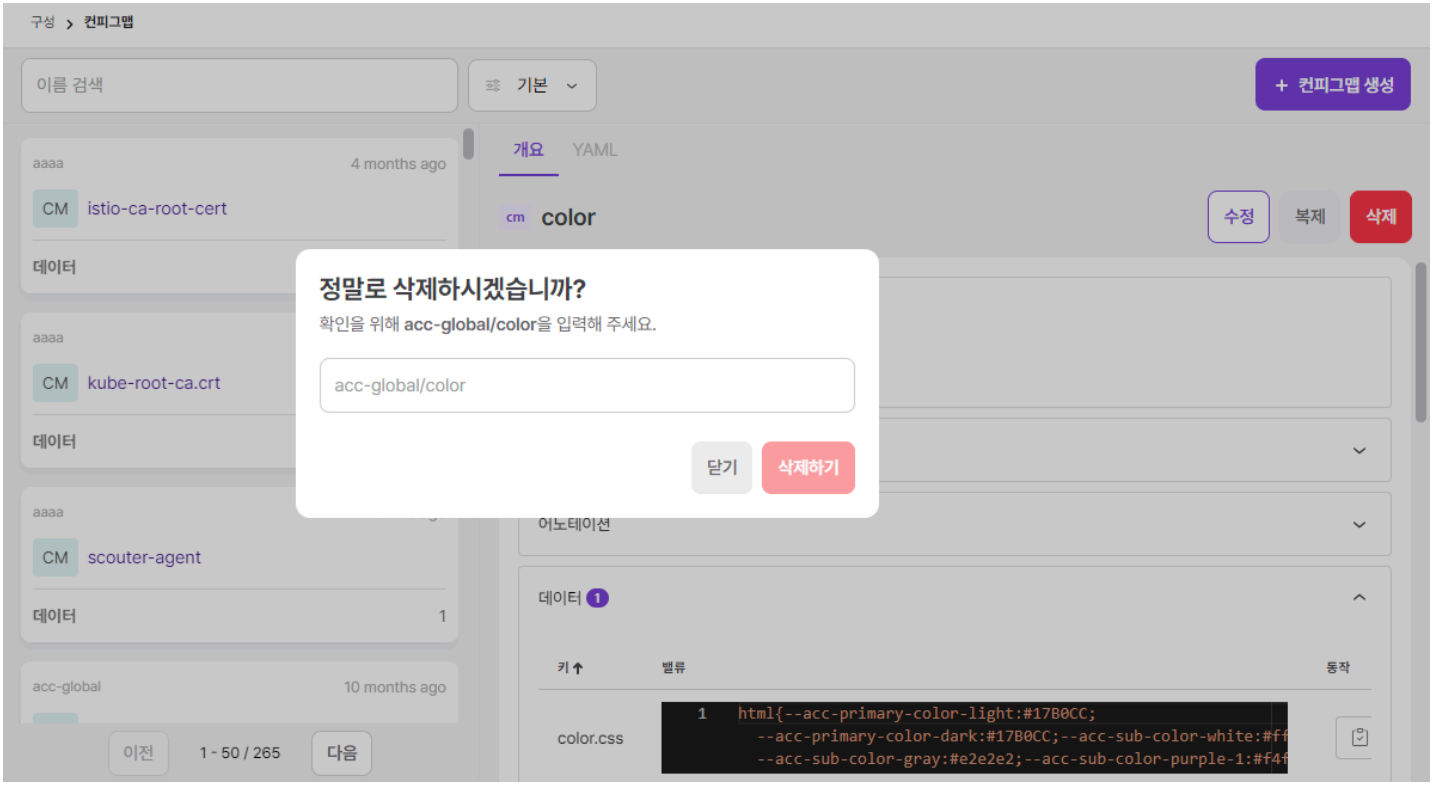
## 컨피그맵 복제

컨피그맵을 다른 클러스터, 네임스페이스에 복제할 수 있다.



## 컨피그맵 삭제

삭제하려는 컨피그맵을 선택하고 우측의 삭제 버튼을 선택한다.




모달에서 네임스페이스와 컨피그맵 이름을 입력하여 삭제한다.

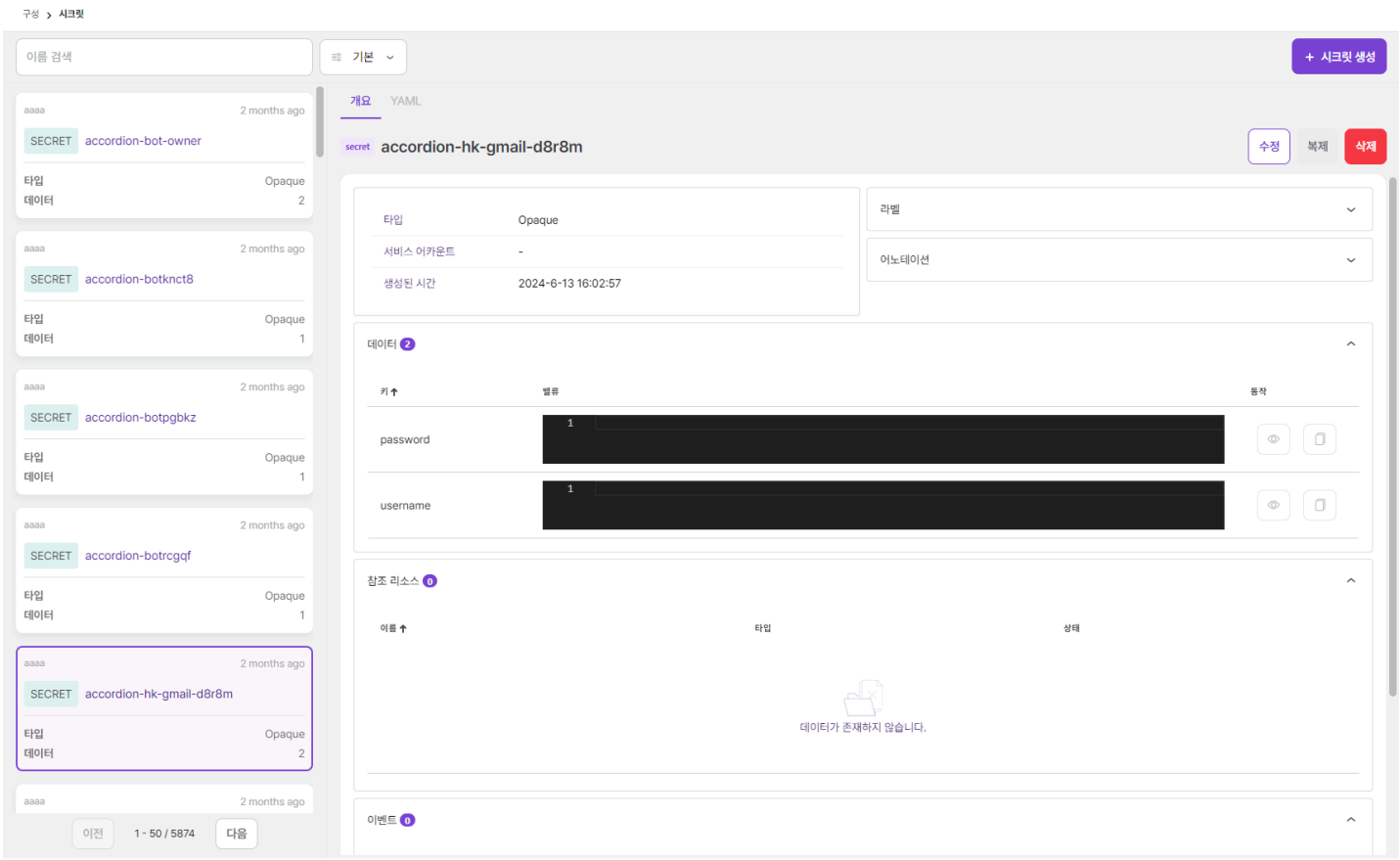
## 4.2.7.2. 시크릿

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.

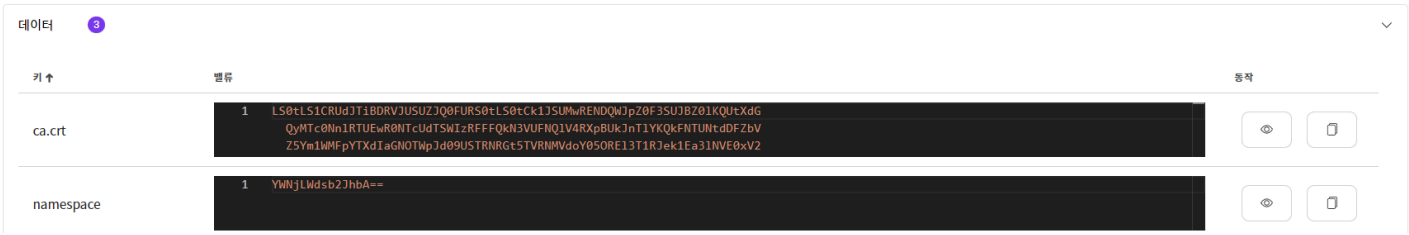
### 개요

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




세부 정보
시크릿에 대한 세부 정보를 제공한다.

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

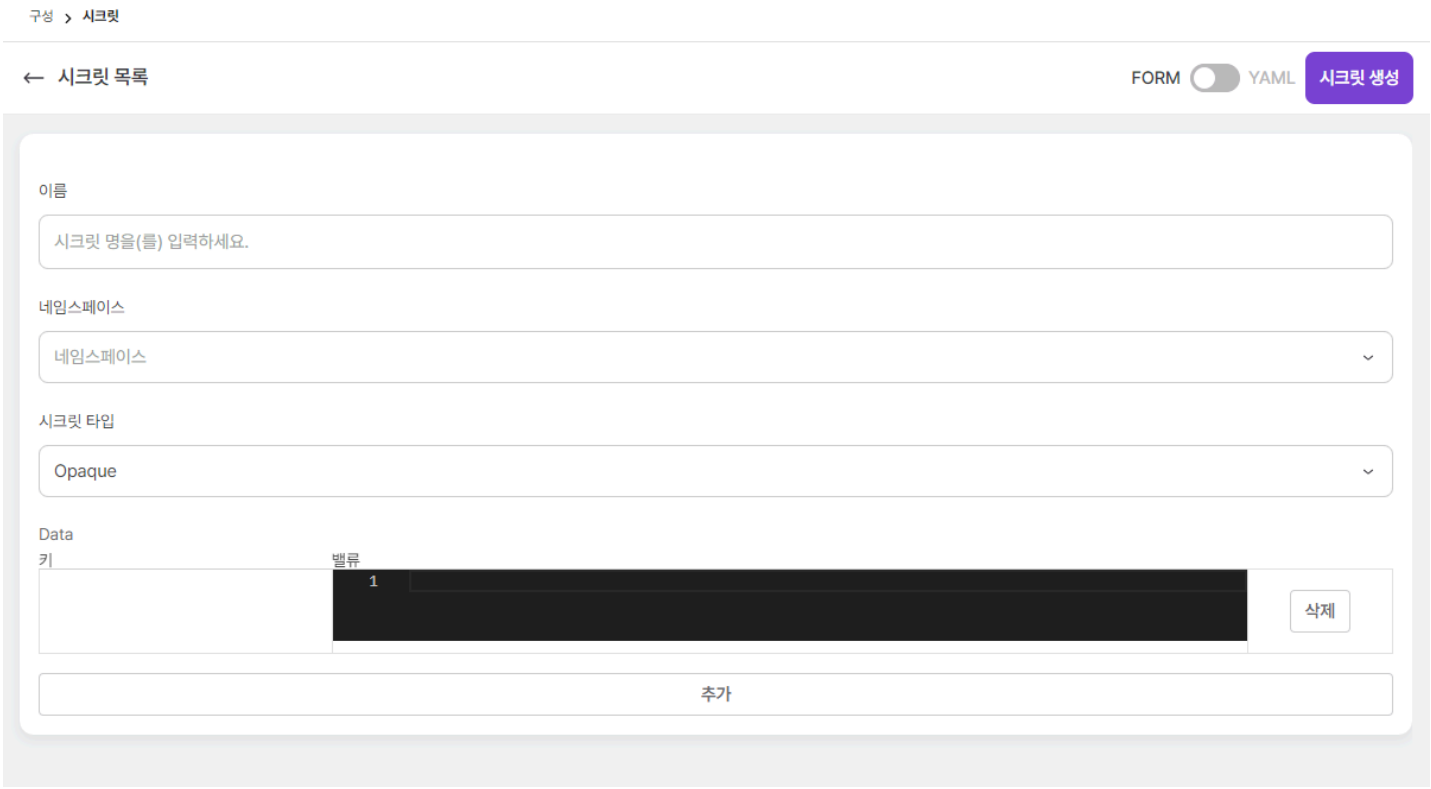



참조 리소스

컨피그맵과 내용이 동일하기 때문에 컨피그맵 참조 리소스를 참고한다.
이벤트 정보
컨피그맵과 내용이 동일하기 때문에 컨피그맵 이벤트 정보를 참고한다.

## 시크릿 생성

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

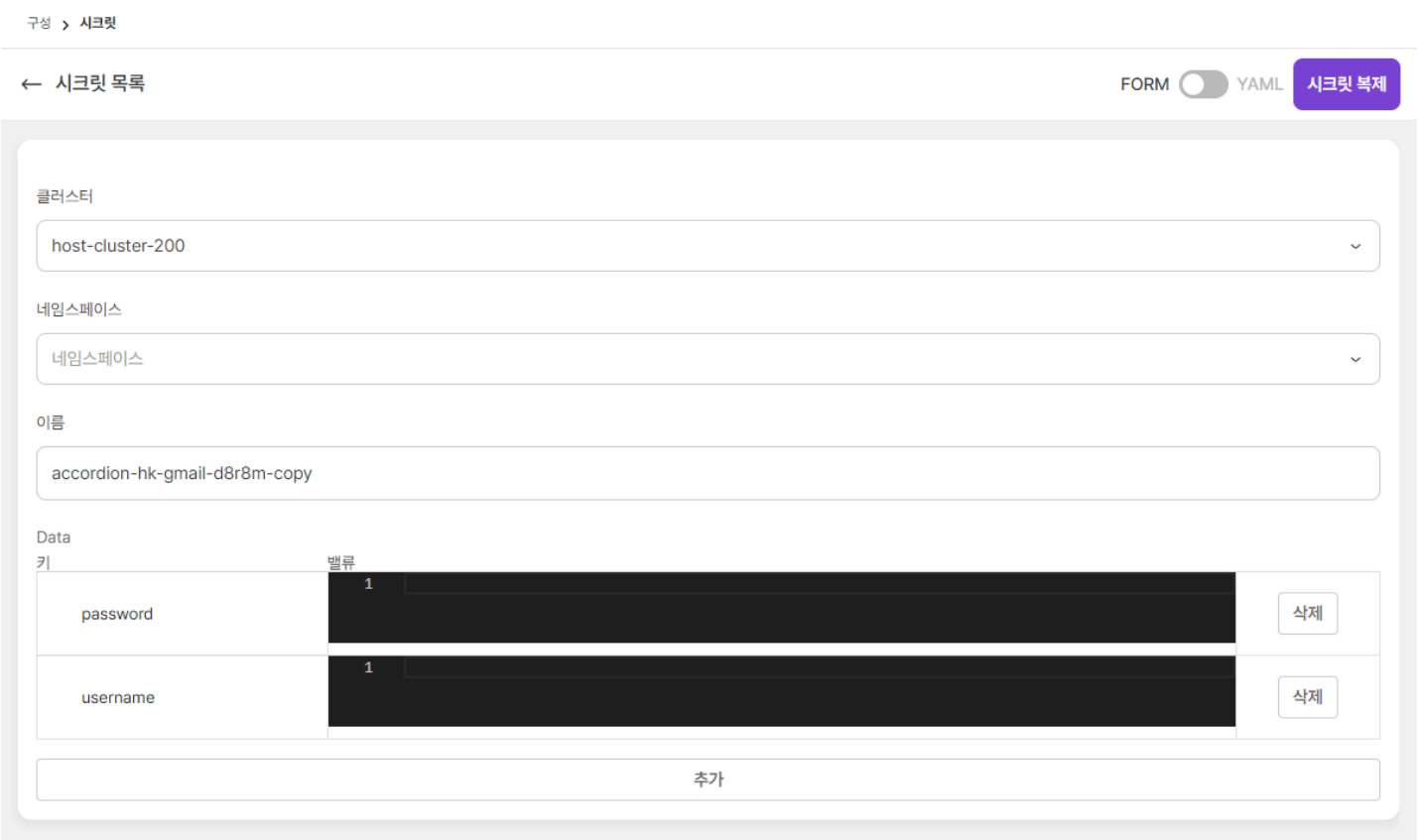



## 시크릿 수정

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

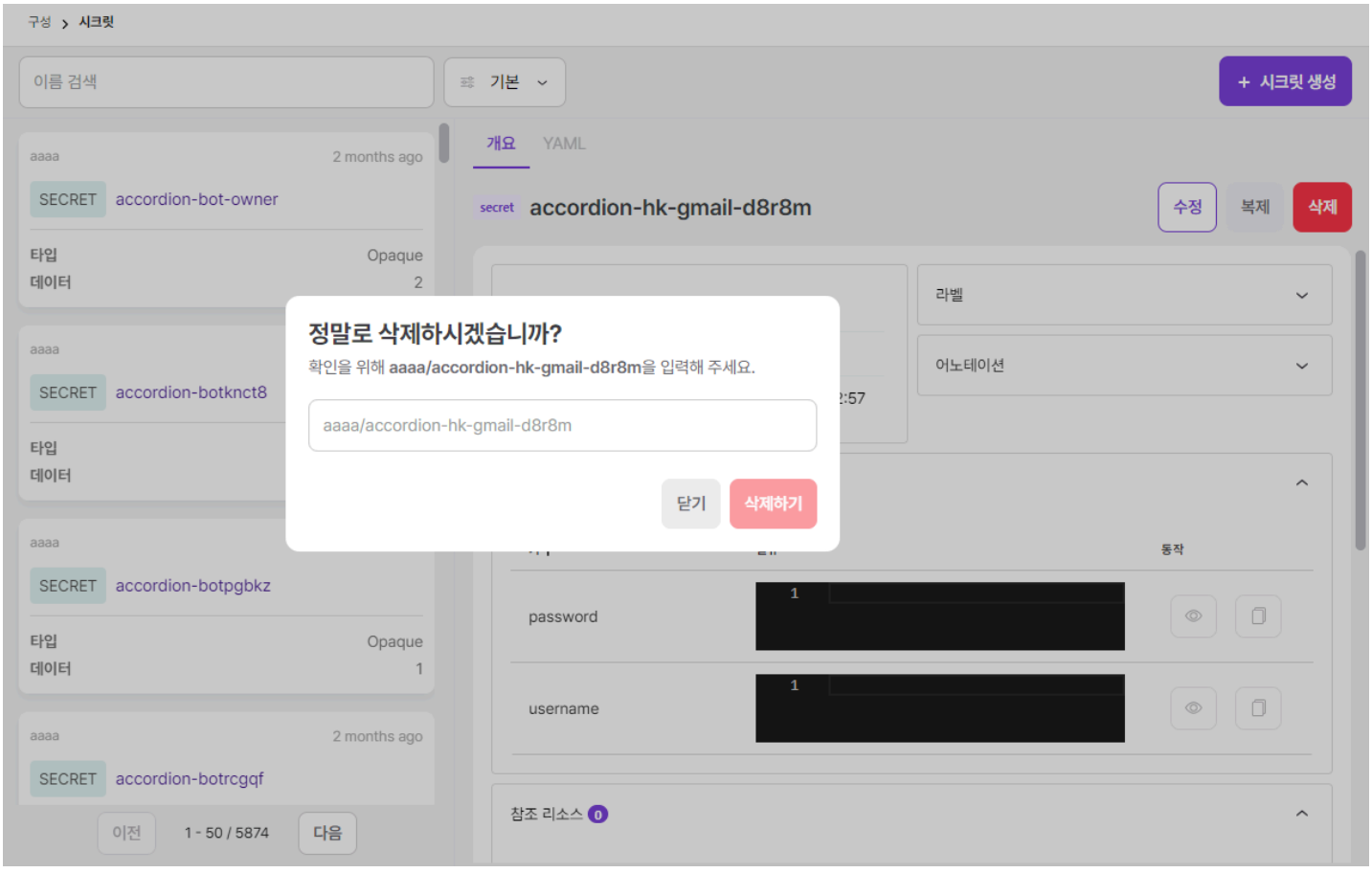
## 시크릿 복제

시크릿을 다른 클러스터, 네임스페이스에 복제할 수 있다.




## 시크릿 삭제

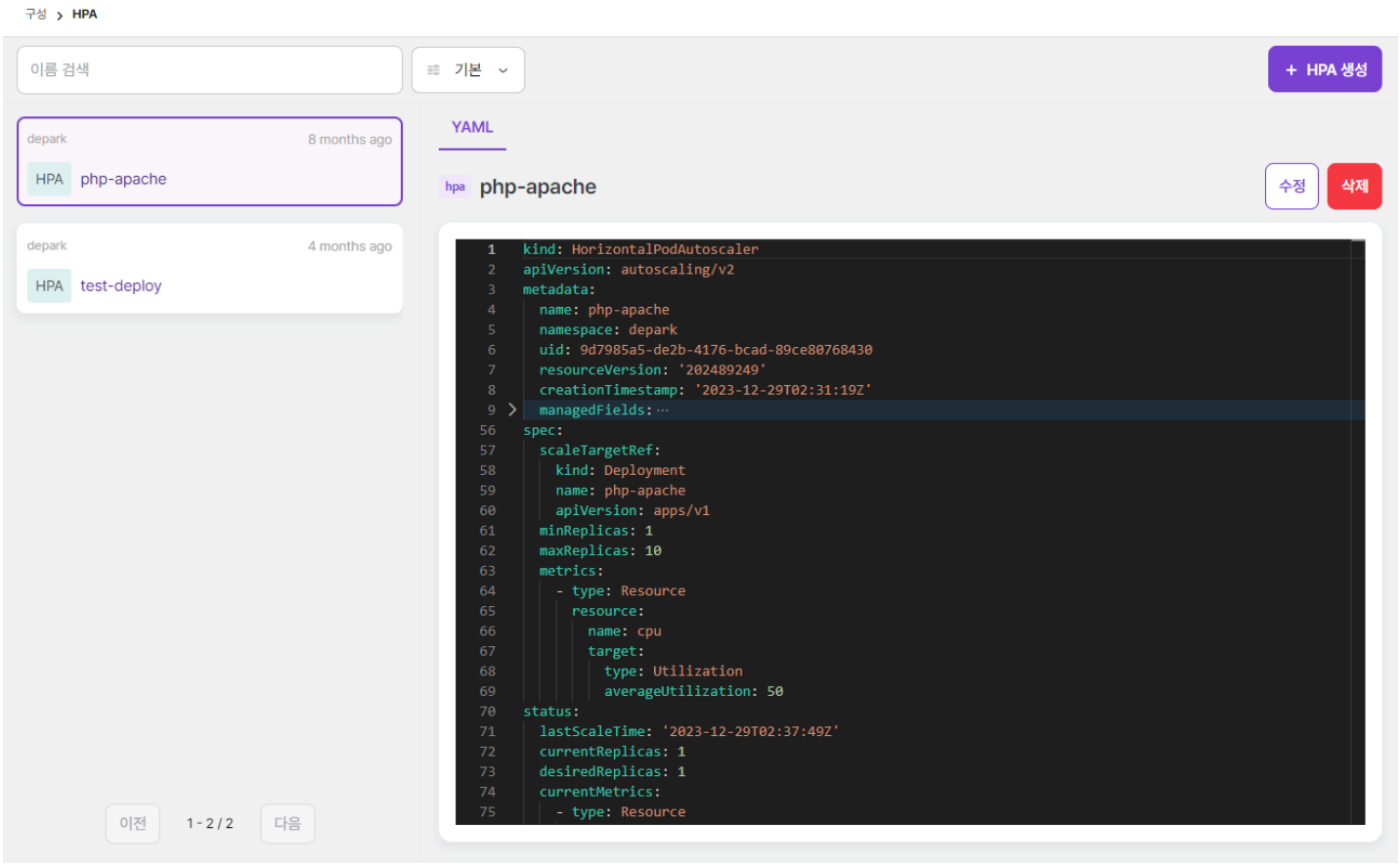
삭제하려는 시크릿을 선택하고 우측의 `삭제` 버튼을 선택한다.



모달에서 네임스페이스와 시크릿 이름을 입력하여 삭제한다.

### 4.2.7.3. HPA

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




#### HPA 생성

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

구성 › HPA

← HPA 목록

HPA 생성

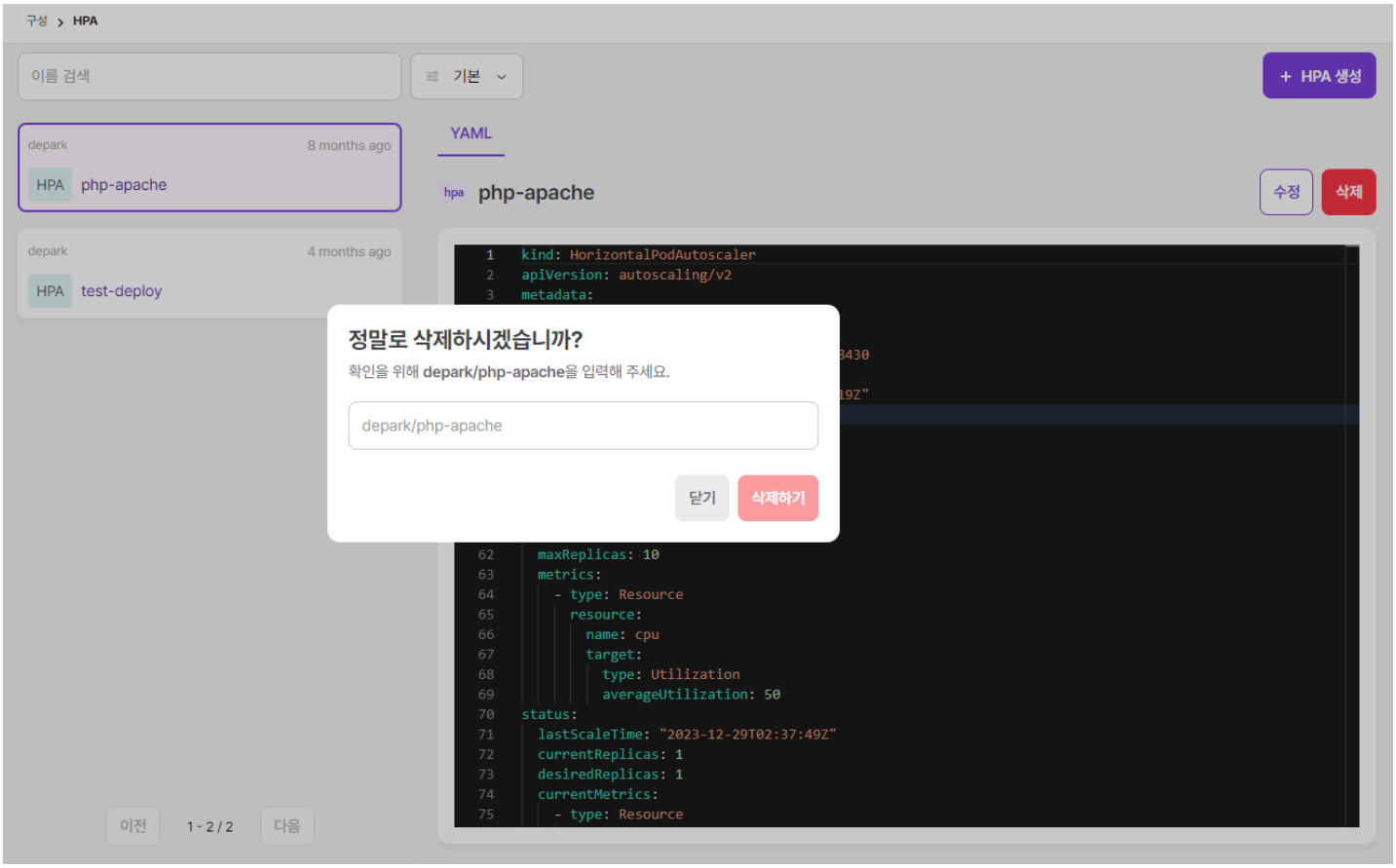
```yaml
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: example-hpa
  namespace: default
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: example-deployment
  minReplicas: 1
  maxReplicas: 10
  metrics:
    - type: Resource
      resource:
        name: cpu
        target:
          type: Utilization
          averageUtilization: 50

```

### HPA 수정

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

### HPA 삭제

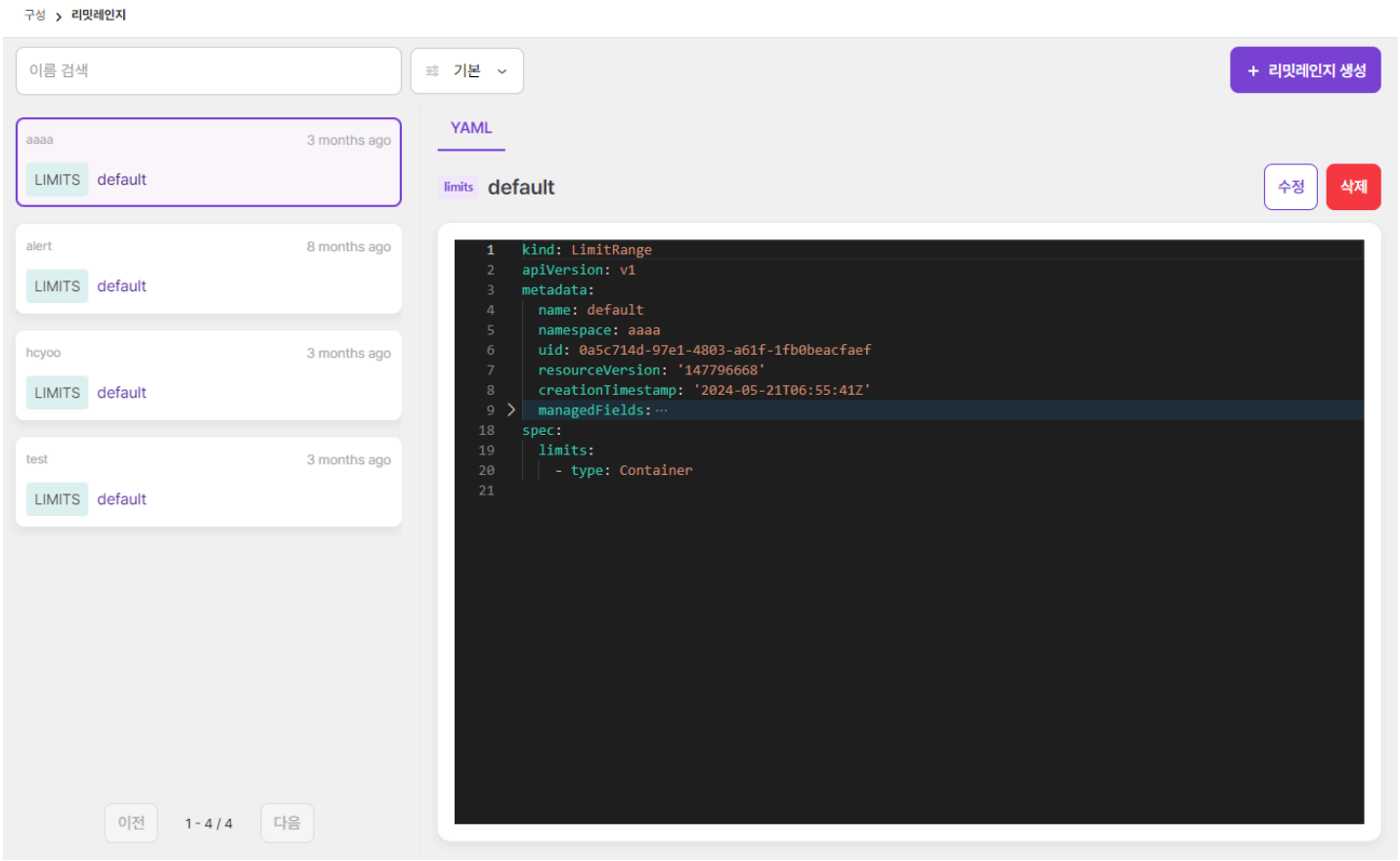
삭제하려는 HPA를 선택하고 우측의 삭제 버튼을 선택한다.




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.

구성 › 리밋레인지

← 리밋레인지 목록

리밋레인지 생성

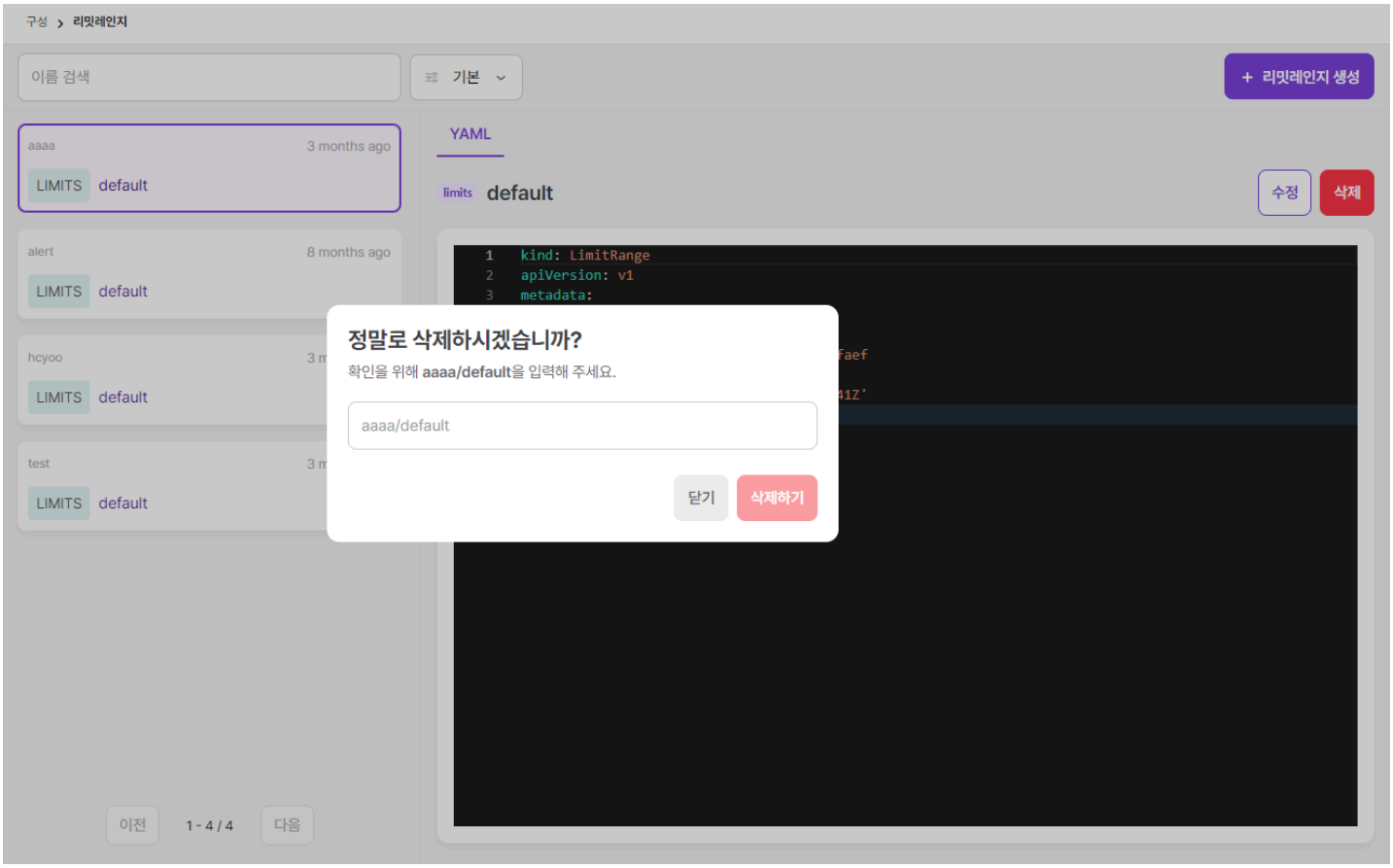
```yaml
apiVersion: v1
kind: LimitRange
metadata:
  name: example-limit-range
  namespace: default
spec:
  limits:
    - default:
        memory:  "512Mi"
      defaultRequest:
        memory: "256Mi"
      max:
        memory: "1Gi"
      min:
        memory: "256Mi"
      type: Container

```

### 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

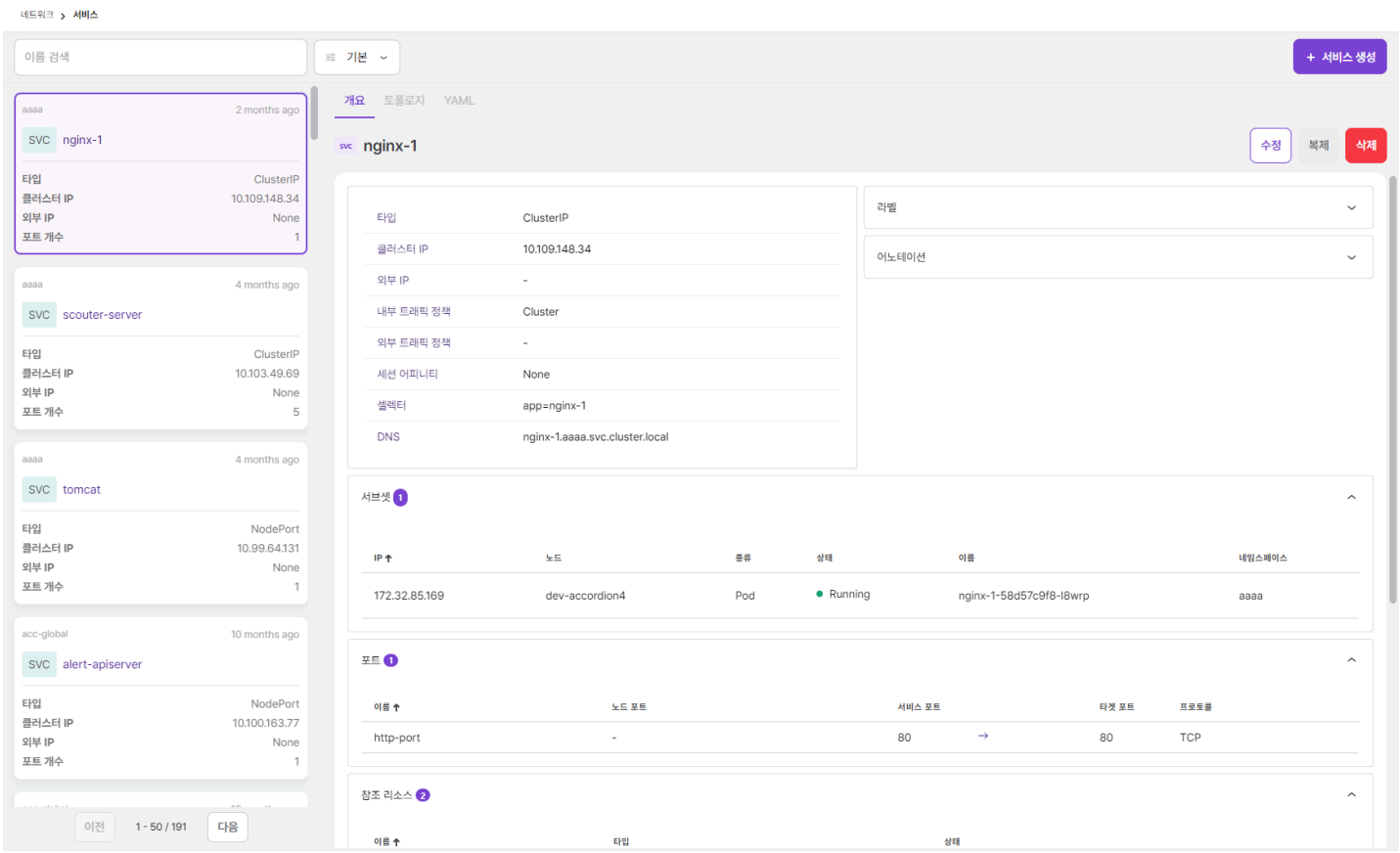
워크로드가 통신하기 위한 네트워크 설정을 관리한다.

### 4.2.8.1. 서비스

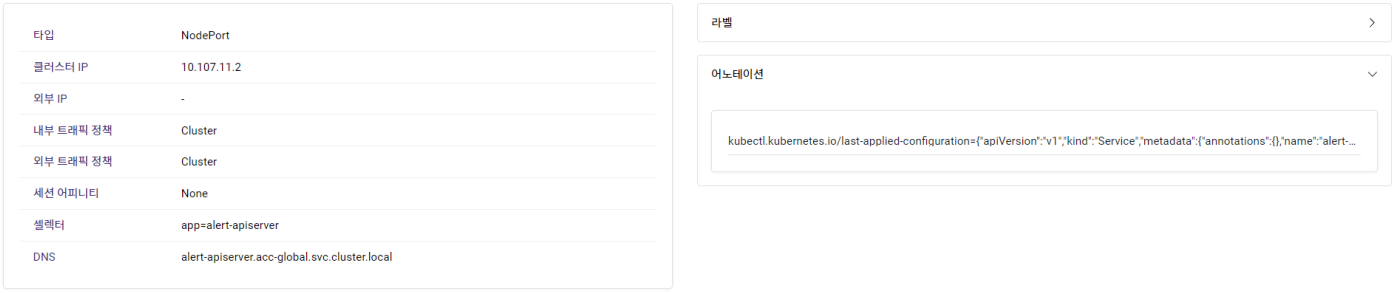
애플리케이션 및 워크로드의 파드 집합에 단일 DNS를 부여하고 로드밸런싱을 수행한다.

#### 개요

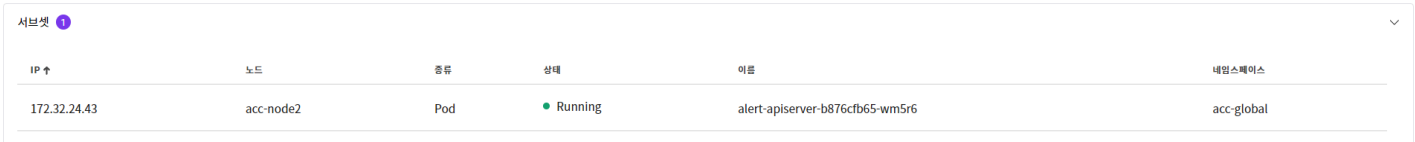
서비스의 세부 정보, 서브셋, 포트, 참조 리소스, 이벤트를 제공한다.




세부 정보
서비스에 대한 세부 정보를 제공한다.

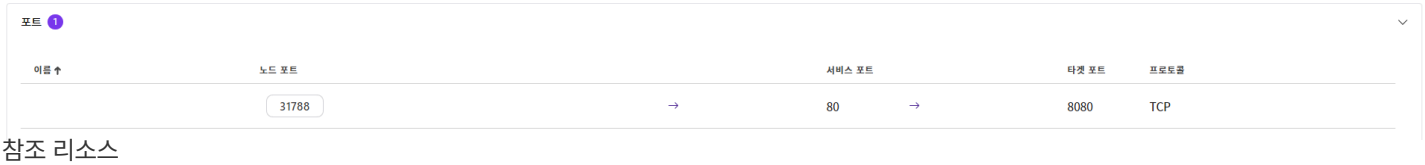



서브셋
서비스에 맵핑된 엔드포인트 정보에 따라 파드 정보를 제공하며 클릭 시 해당 리소스로 이동된다.

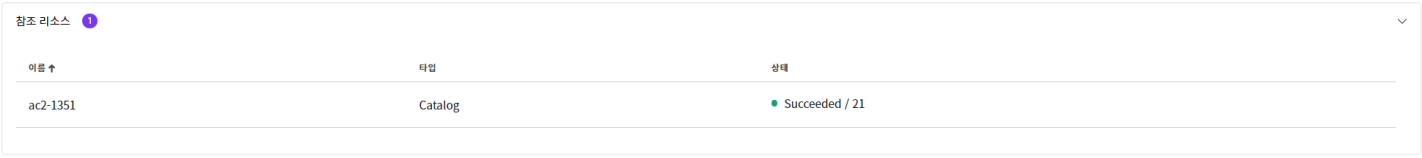



포트

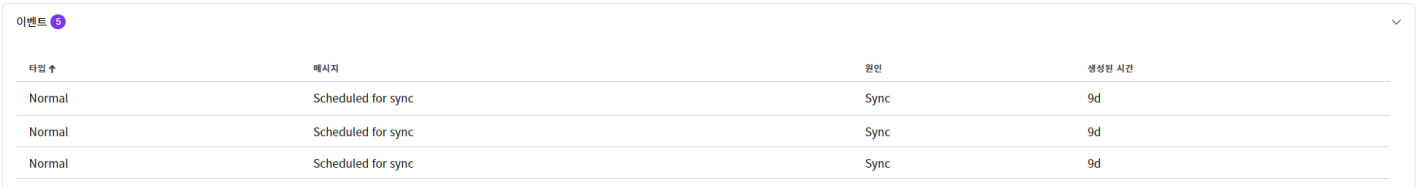
서비스의 포트 정보 및 노드 포트로 접속할 수 있는 버튼을 제공한다.




참조 리소스
서비스의 참조 리소스 정보를 제공하며 클릭 시 해당 리소스로 이동된다.




이벤트 정보
서비스에서 발생하는 이벤트 정보를 제공한다.

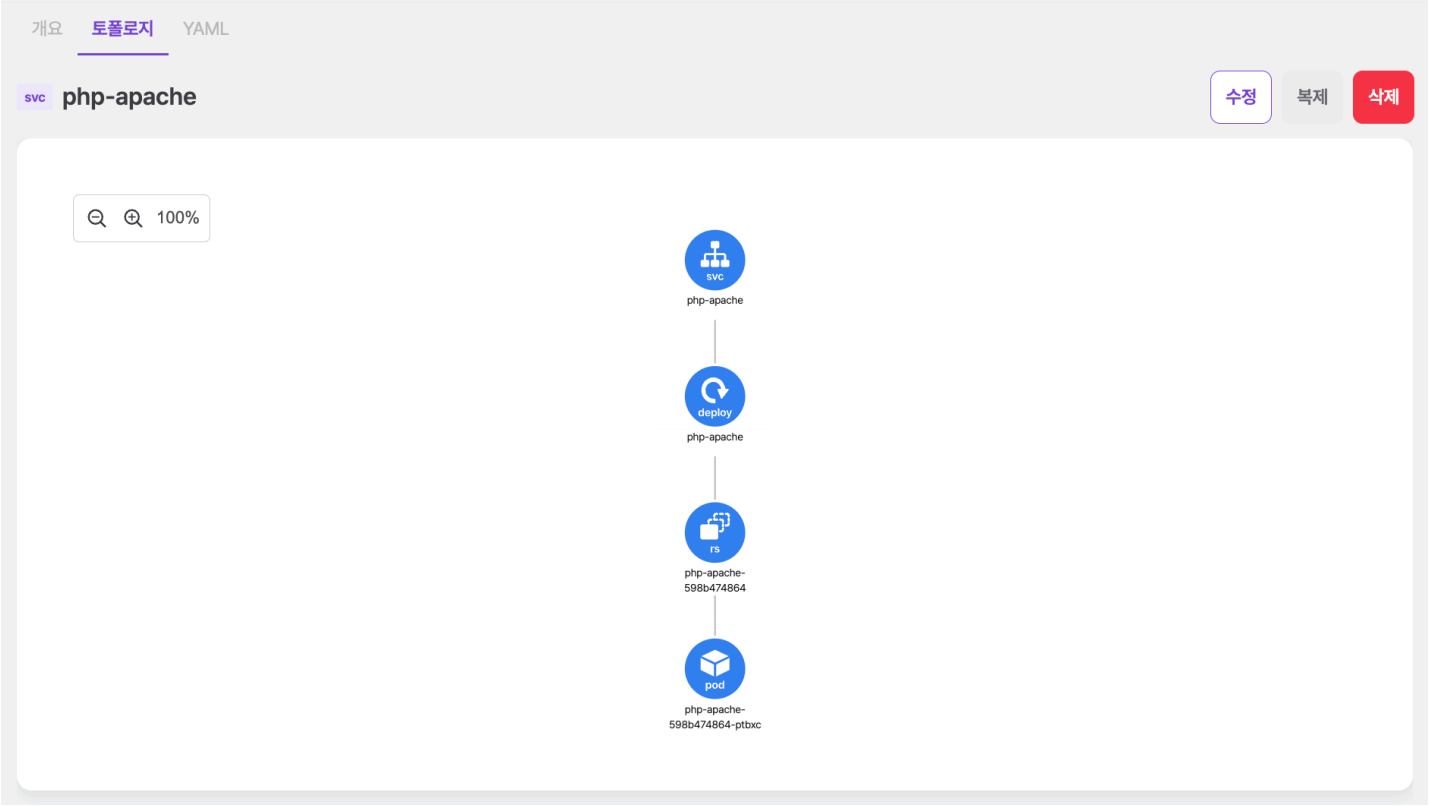



| 항목 | 설명 |
| --- | --- |
| 타입 | 이벤트 타입<br>• Normal: 일반적인 작업으로 발생된 이벤트<br>• Warning: 오류에 의해 발생된 이벤트 |
| 메시지 | 이벤트 메시지 |
| 원인 | 이벤트 발생 이유 |
| 생성된 시간 | 이벤트 생성 후 지난 시간 |

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

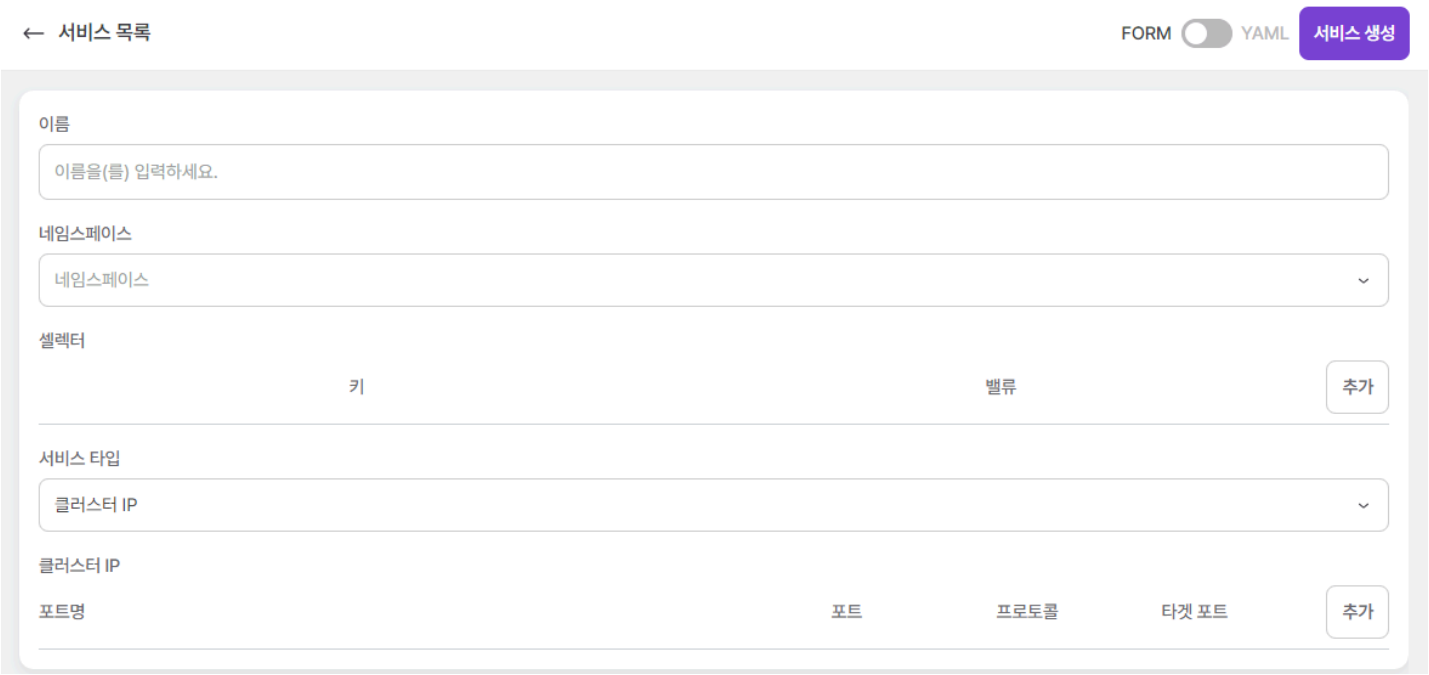
서비스의 토폴로지에서 출력되는 리소스는 다음과 같다.

- 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.

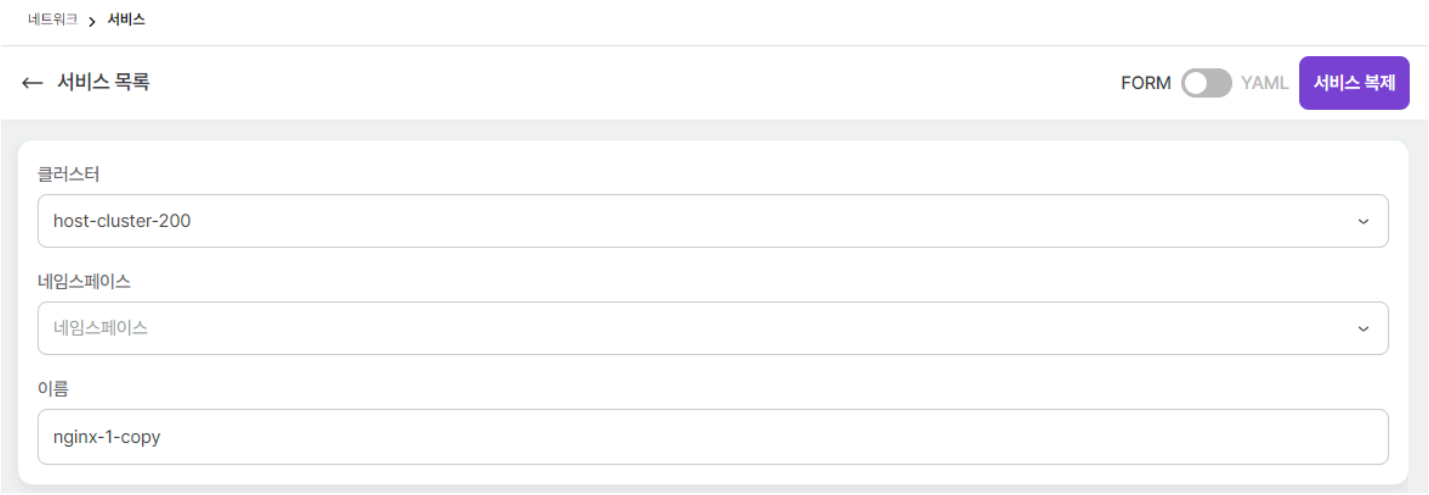


## 서비스 수정

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

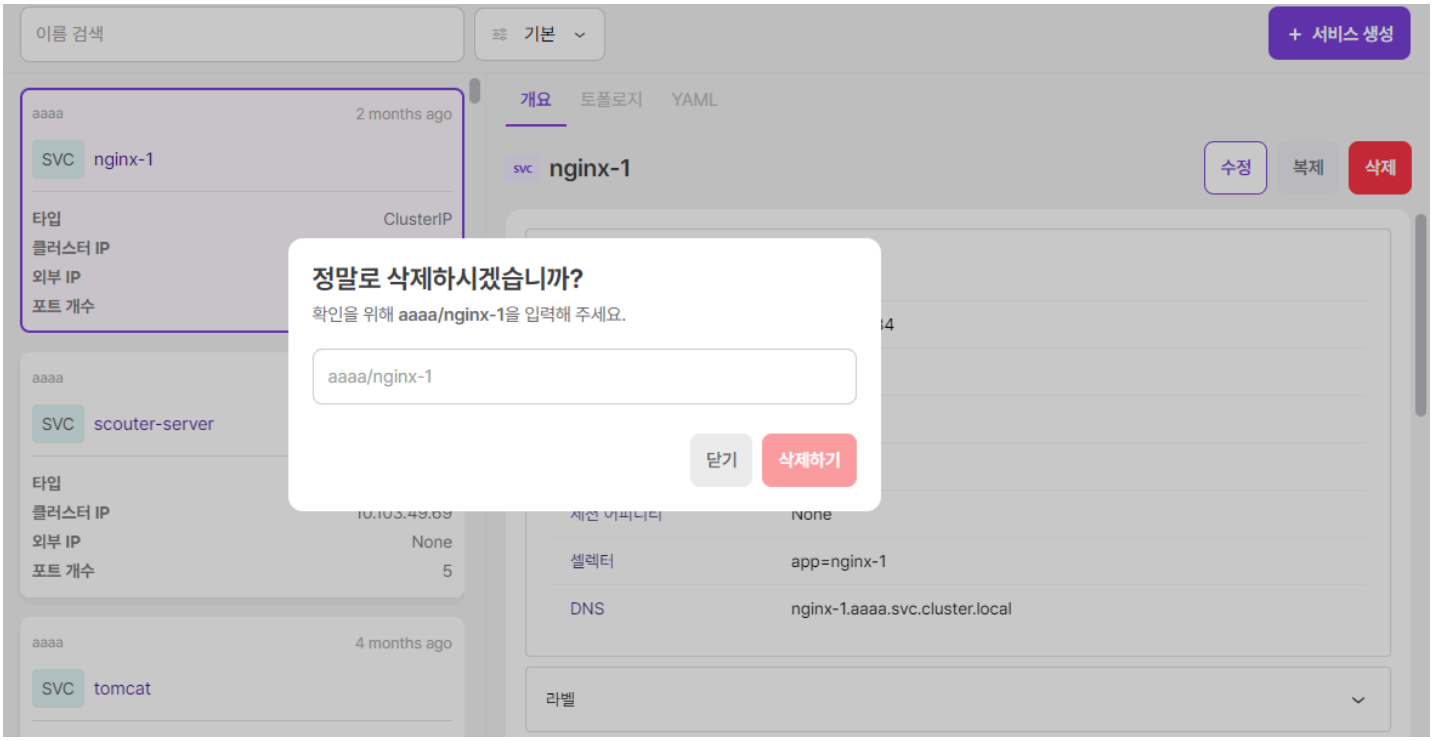
## 서비스 복제

서비스를 다른 클러스터, 네임스페이스에 복제할 수 있다.




## 서비스 삭제

삭제하려는 서비스를 선택하고 우측의 삭제 버튼을 선택한다.



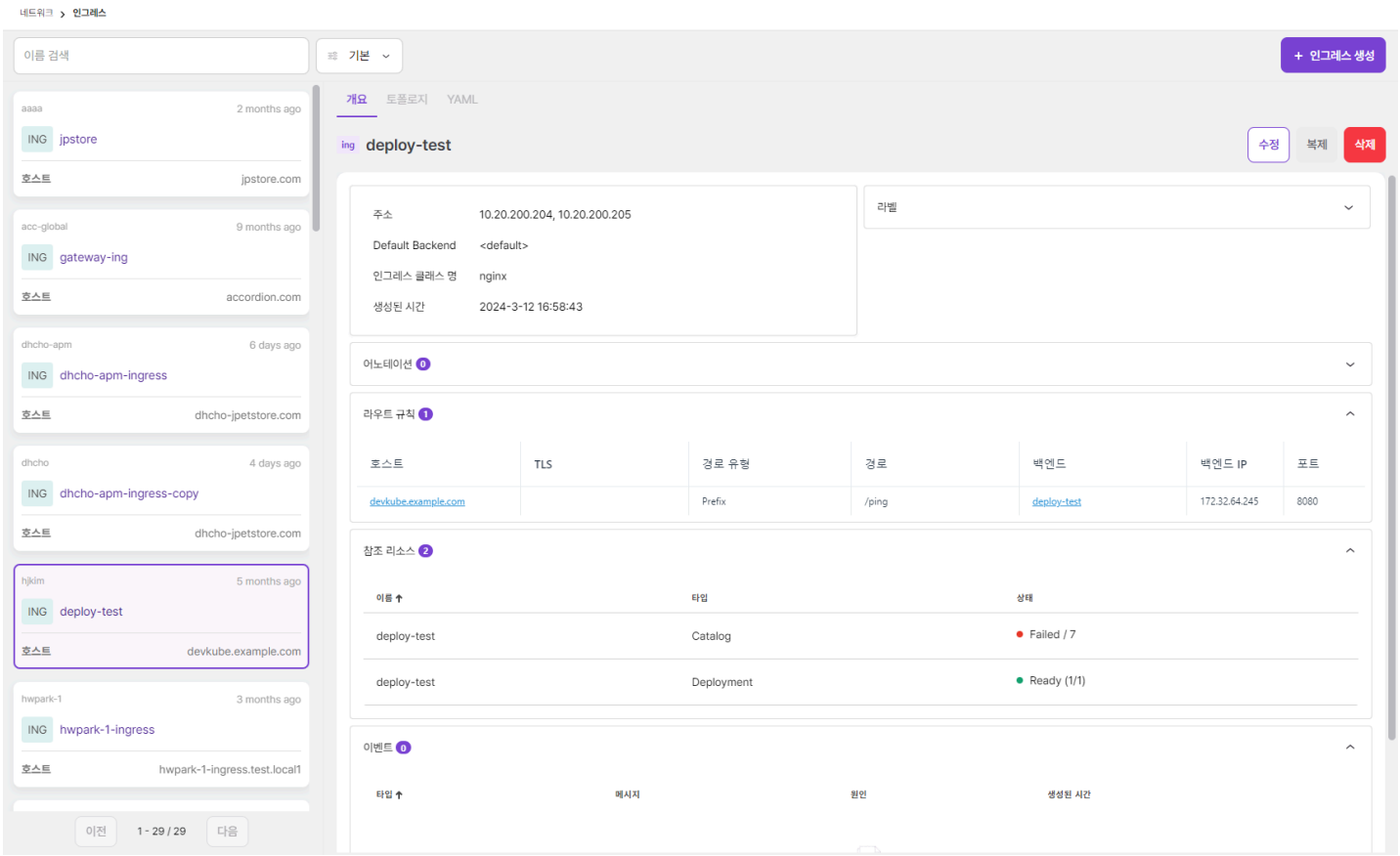

모달에서 네임스페이스와 서비스 이름을 입력하여 삭제한다.

## 4.2.8.2. 인그레스

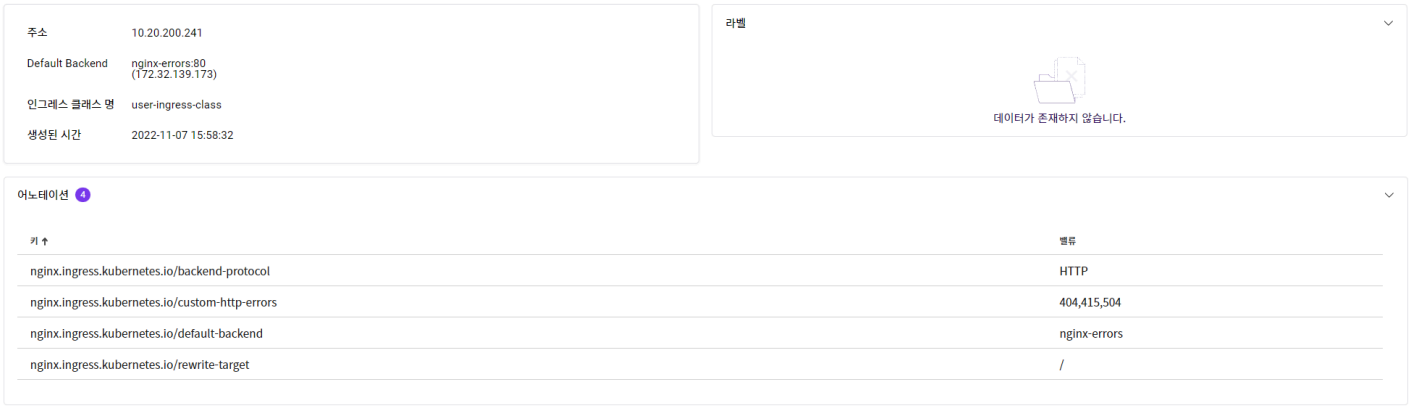
인그레스는 클러스터 외부에서 클러스터 내부 서비스로 HTTP와 HTTPS 경로를 노출한다. 인그레스는 외부에서 서비스로 접속이 가능한 URL, 로드 밸런스 트래픽, SSL/TLS 종료 그리고 이름-기반의 가상 호스팅을 제공하도록 구성할 수 있다.

### 개요

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




세부 정보
인그레스에 대한 세부 정보를 제공한다.



라우트 규칙

| 항목 | 설명 |
| --- | --- |
| 호스트 | 네트워크 요청이 들어오는 호스트 명 |
| TLS | https 설정을 위한 시크릿 정보 |
| 경로 유형 | • ImplementationSpecific : IngressClass의 설정에 따름<br>• Exact : url경로를 정확하게 일치<br>• Prefix : /로 분할된 url 경로 접두사를 기반으로 일치 |
| 경로 | 요청의 경로 |
| 백엔드 | 호스트와 경로에 연결된 서비스 |
| 백엔드 IP | 서비스의 IP |
| 포트 | 서비스의 포트 |

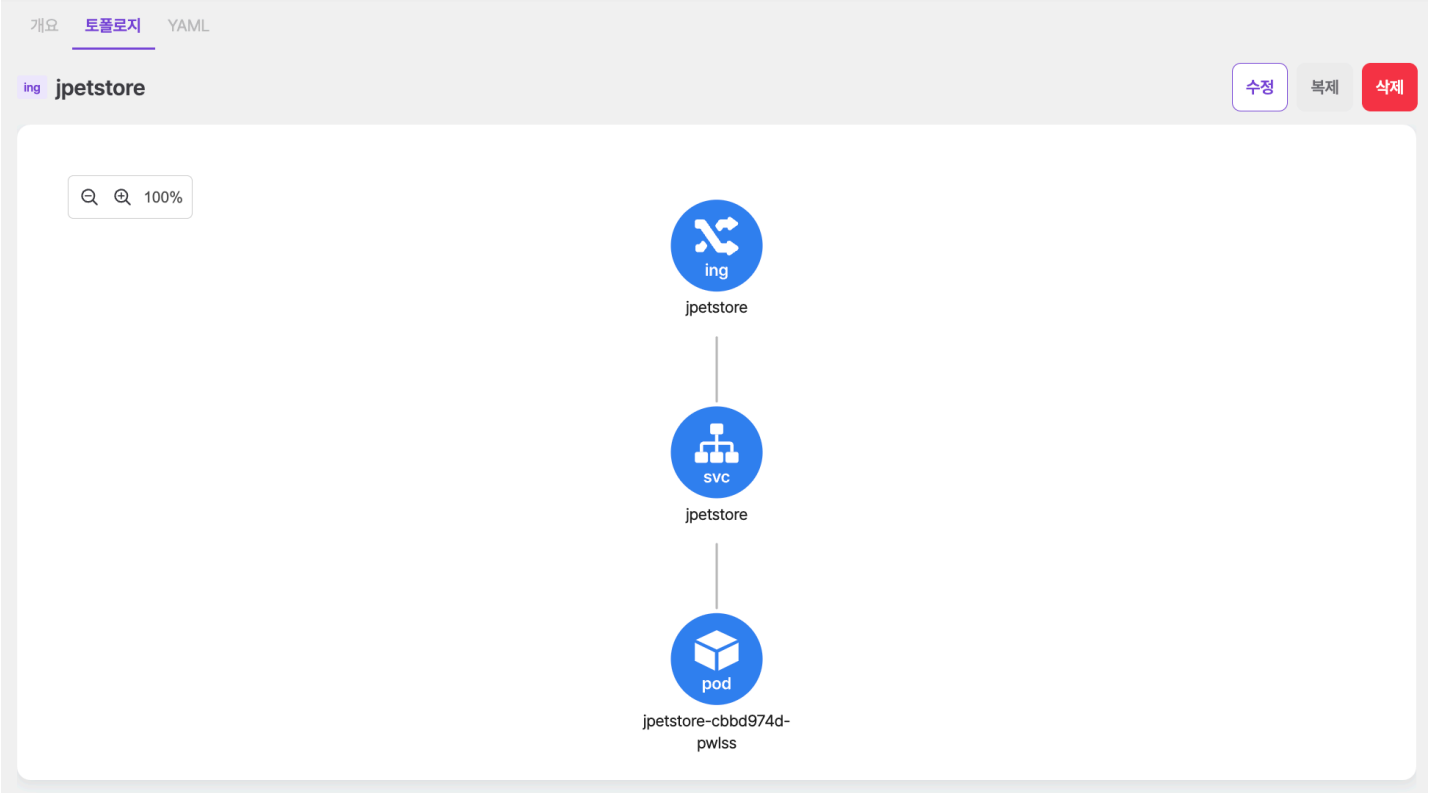
참조 리소스
서비스와 내용이 동일하기 때문에 서비스 참조 리소스를 참고한다.

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




## 인그레스 생성

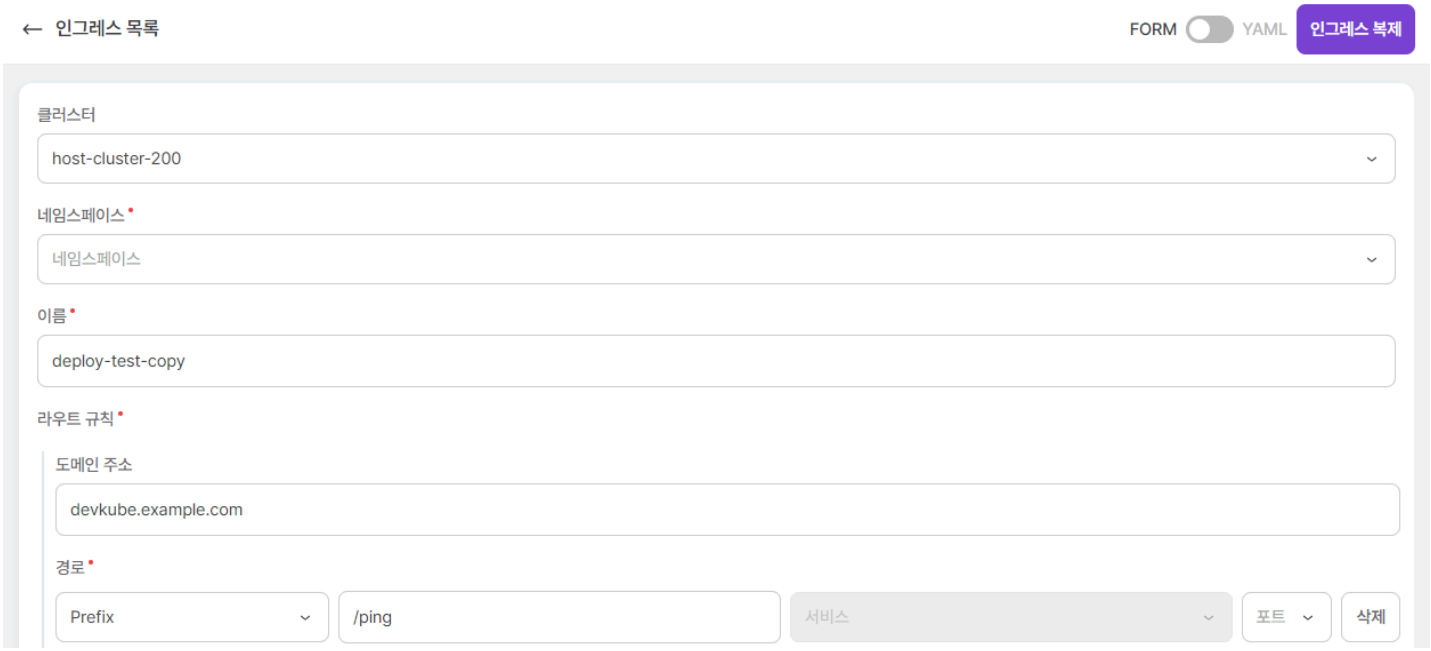
+ 인그레스 생성 을 선택하면 나타나는 화면에서 쿠버네티스 인그레스 리소스 정보를 입력하여 생성할 수 있다. 생성 시에는 FORM/YAML로 입력할 수 있다.

## 인그레스 수정

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

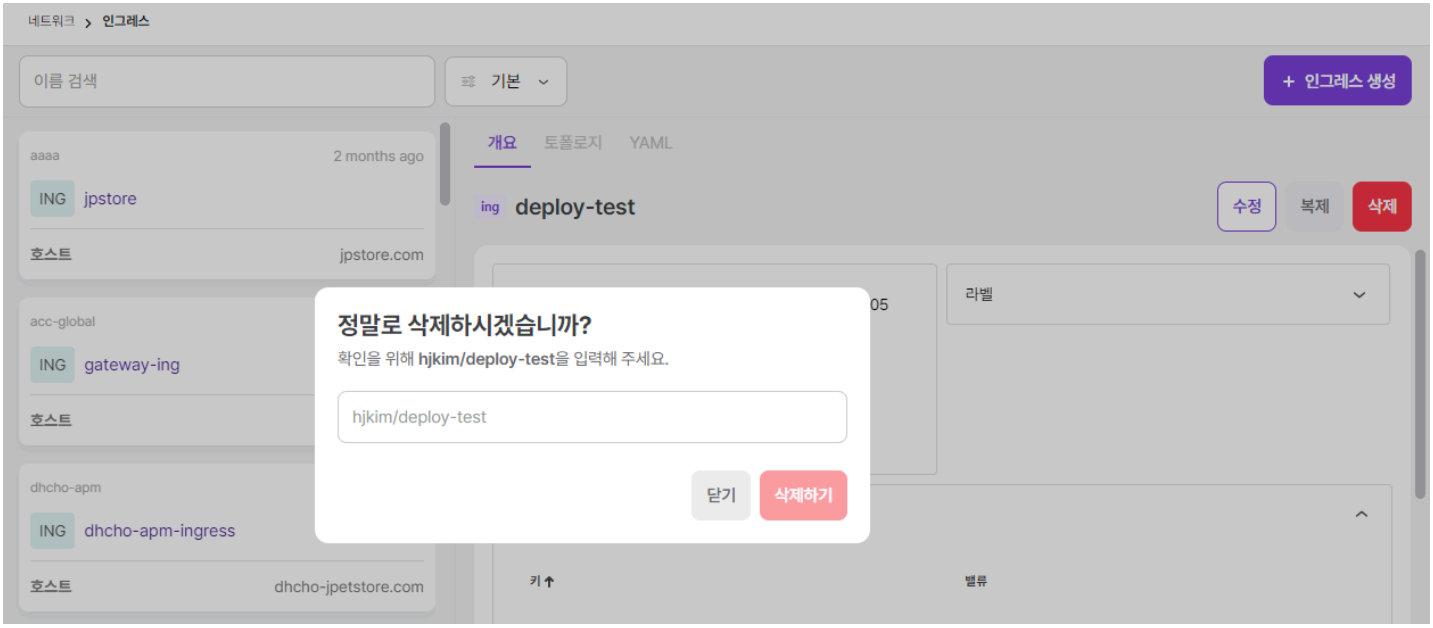
## 인그레스 복제

인그레스를 다른 클러스터, 네임스페이스에 복제할 수 있다.




## 인그레스 삭제

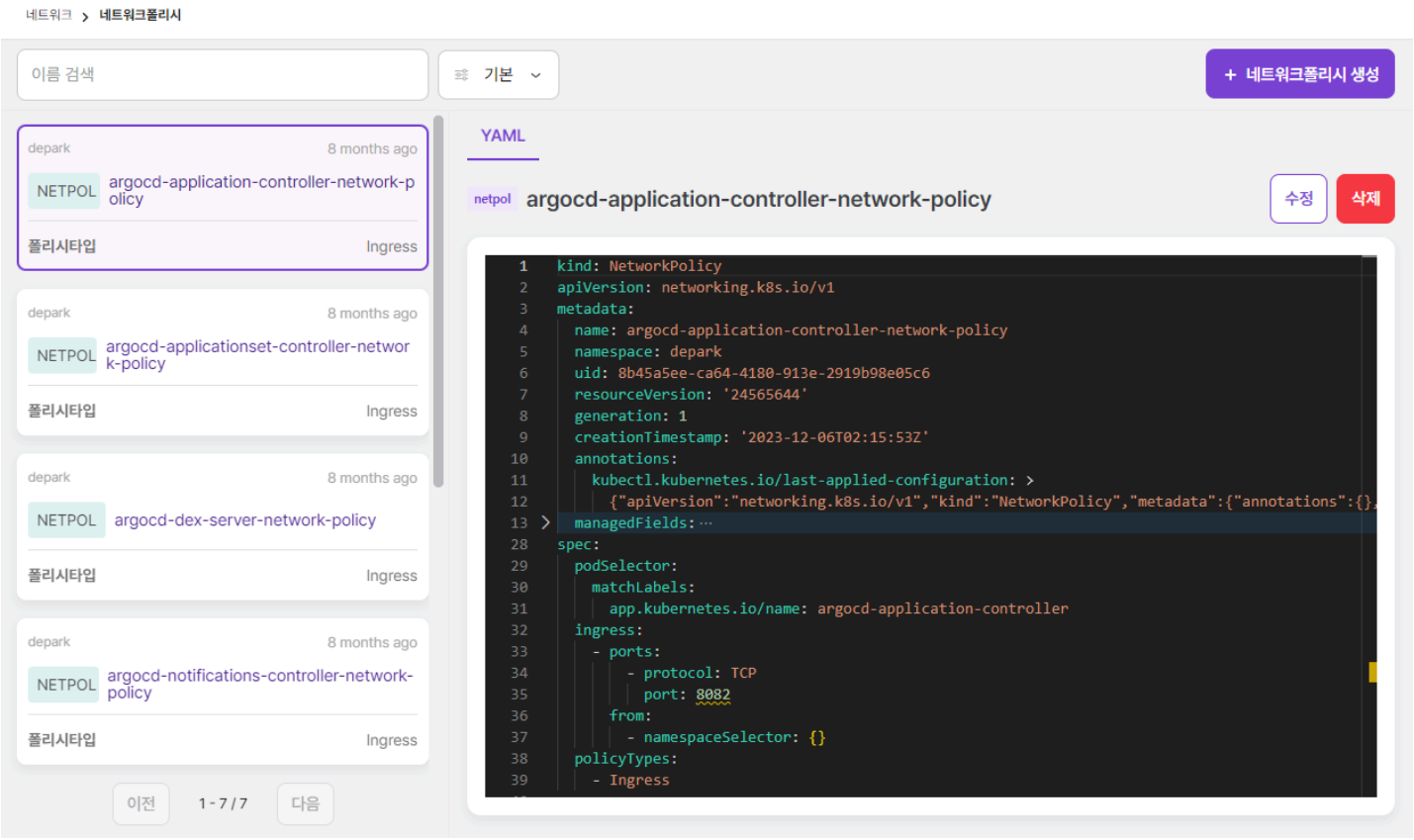
삭제하려는 인그레스를 선택하고 우측의 `삭제` 버튼을 선택한다.




모달에서 네임스페이스와 인그레스 이름을 입력하여 삭제한다.

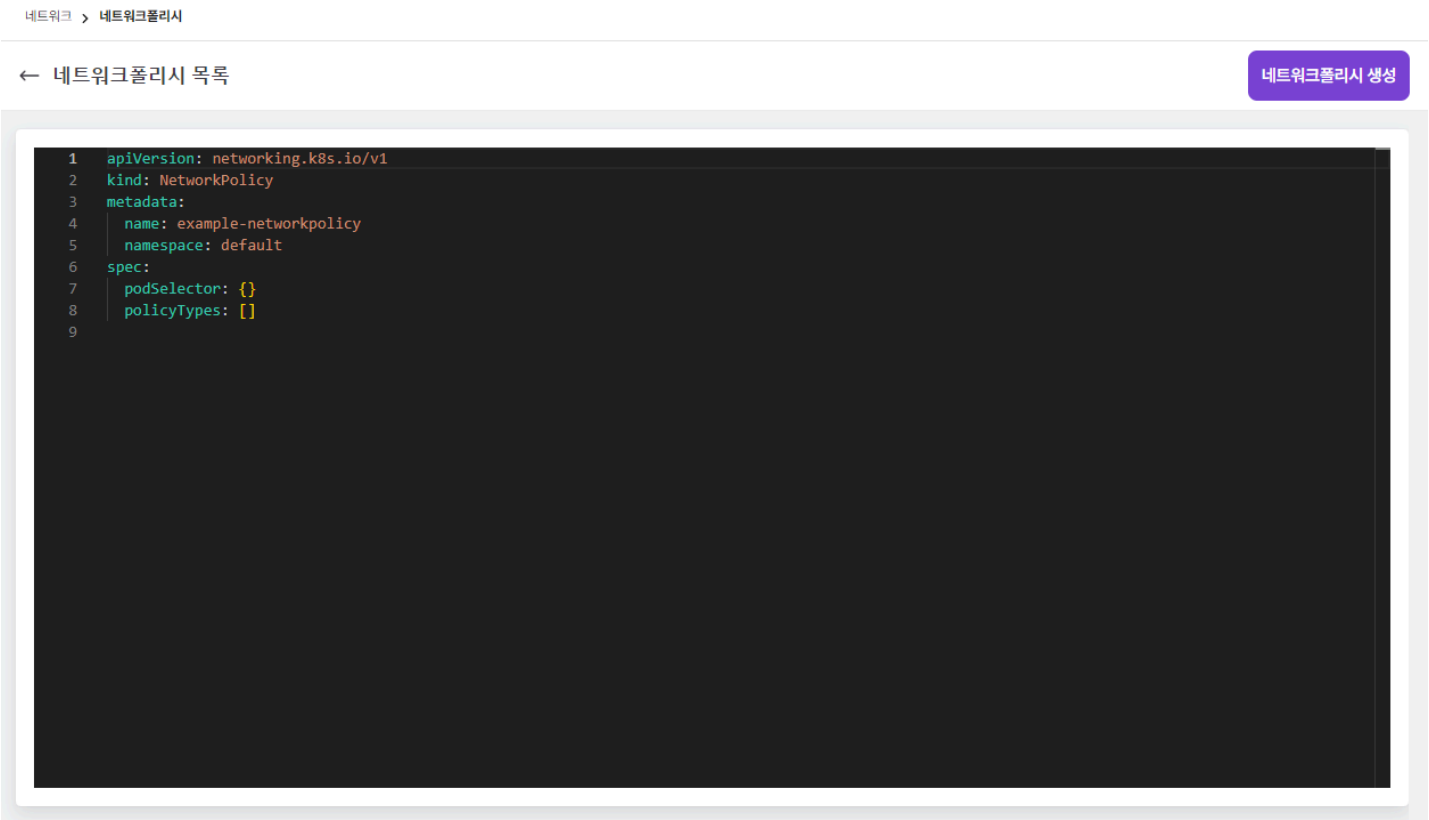
## 4.2.8.3. 네트워크폴리시

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




### 네트워크폴리시 생성

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

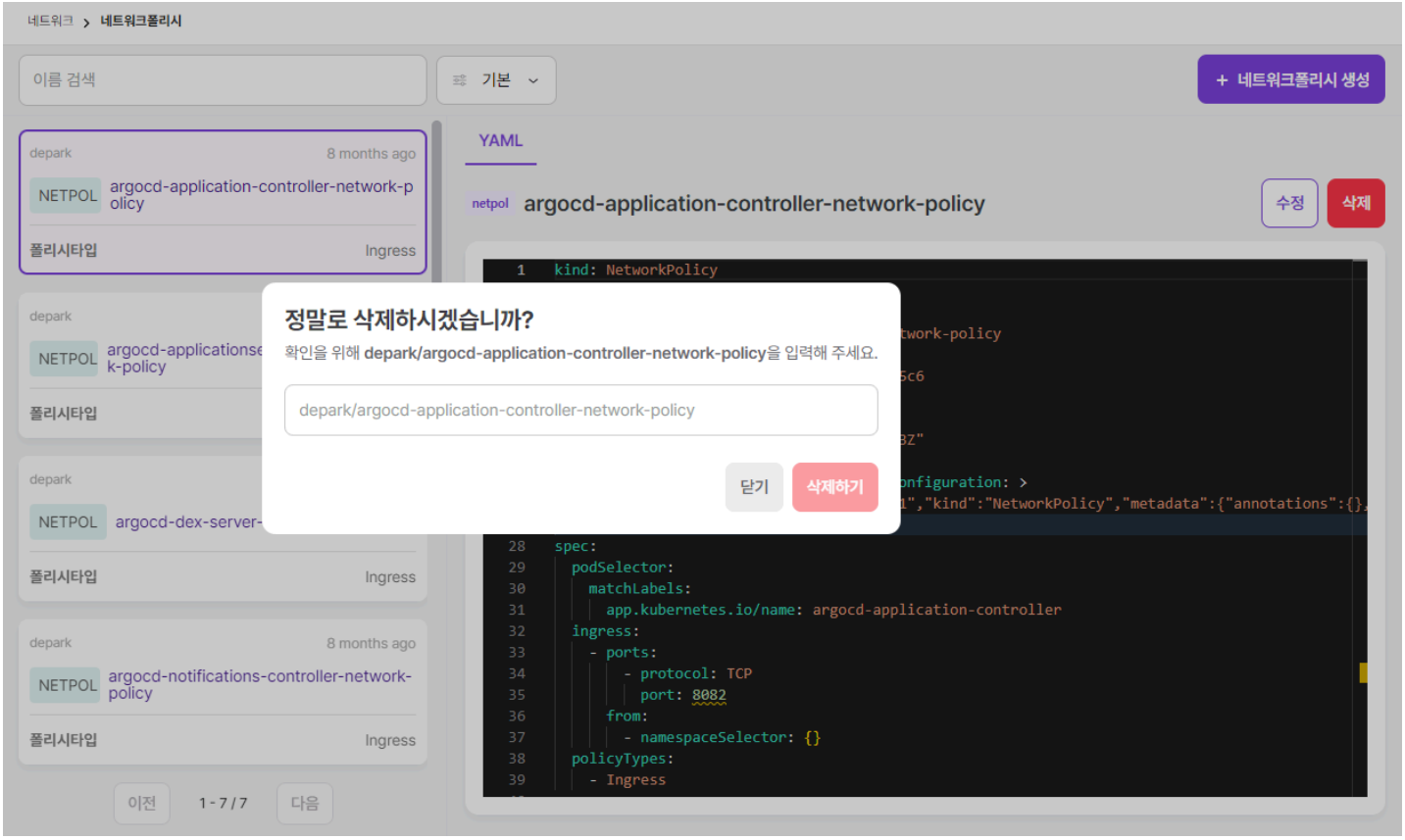



### 네트워크폴리시 수정

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

### 네트워크폴리시 삭제

삭제하려는 네트워크폴리시를 선택하고 우측의 `삭제` 버튼을 선택한다.



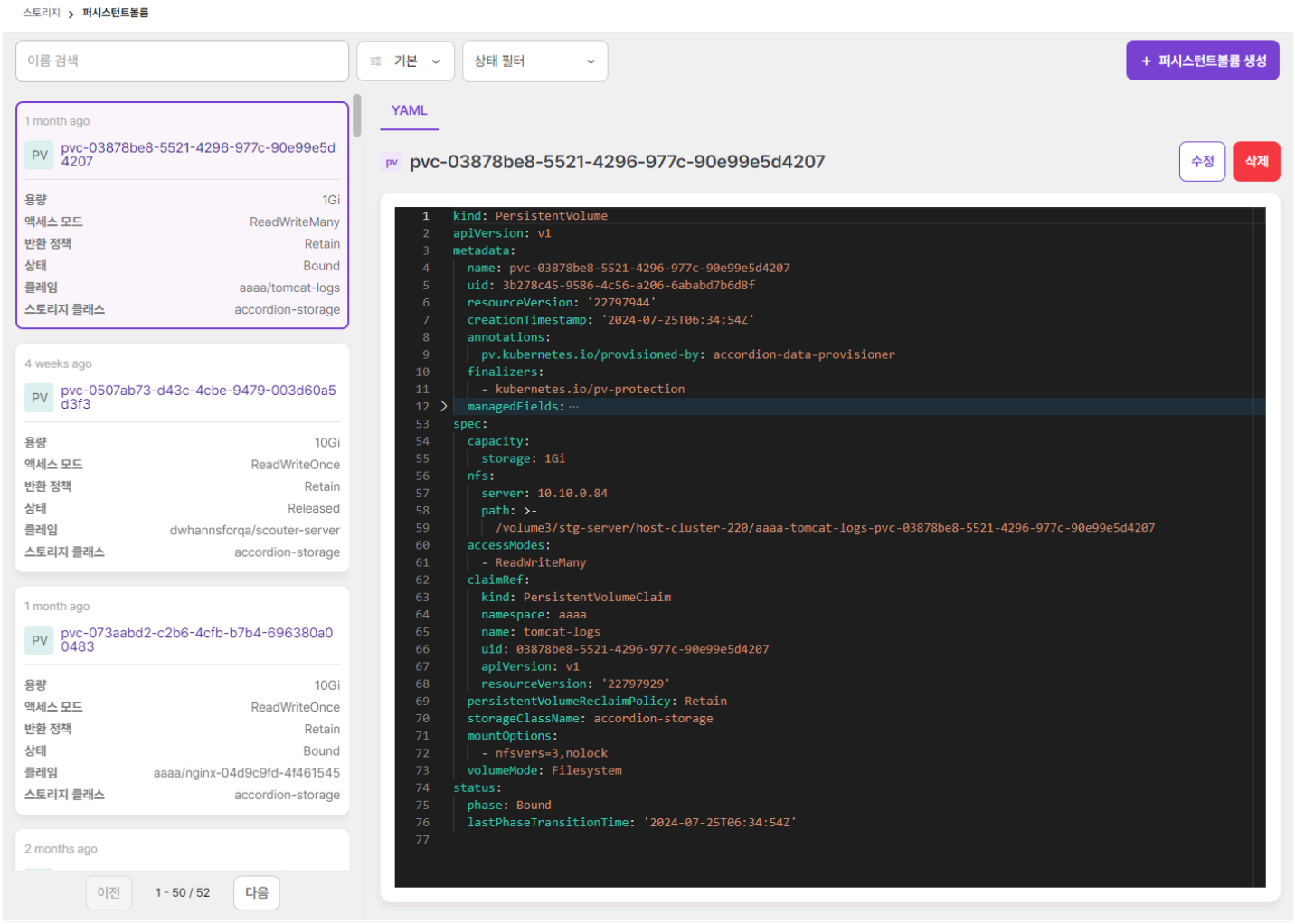

모달에서 네임스페이스와 네트워크폴리시 이름을 입력하여 삭제한다.

## 4.2.9. 스토리지

컨테이너 환경에서 디스크는 일반 환경과 다른 특성을 가지고 있다. 컨테이너 내부에서 저장한 파일은 임시적인 것으로 컨테이너가 삭제되면 저장한 컨텐츠도 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.

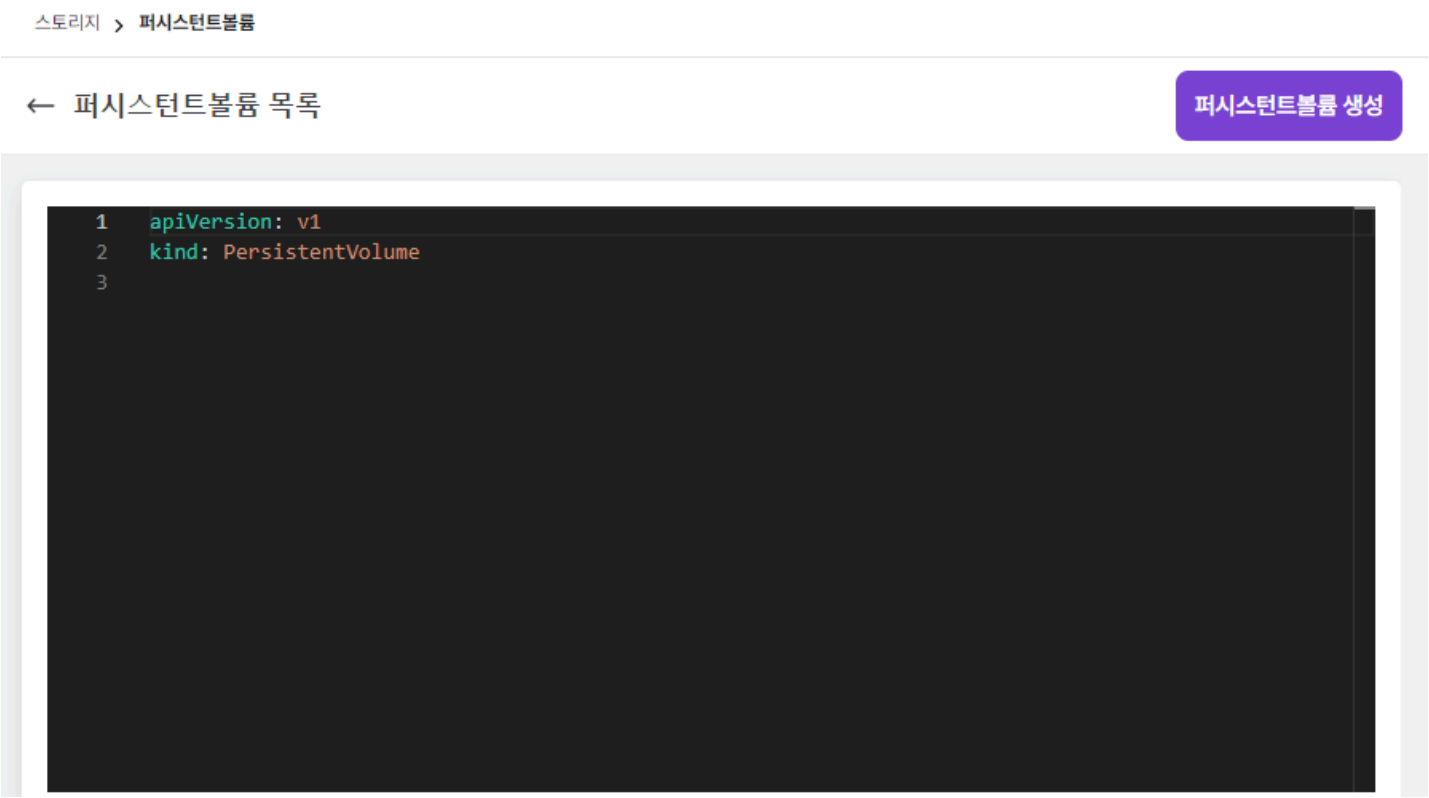
### 4.2.9.1. 퍼시스턴트볼륨

퍼시스턴트볼륨은 프로비저닝하거나 스토리지 클래스를 사용하여 동적으로 프로비저닝한 클러스터의 스토리지이다.



### 퍼시스턴트볼륨 생성

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

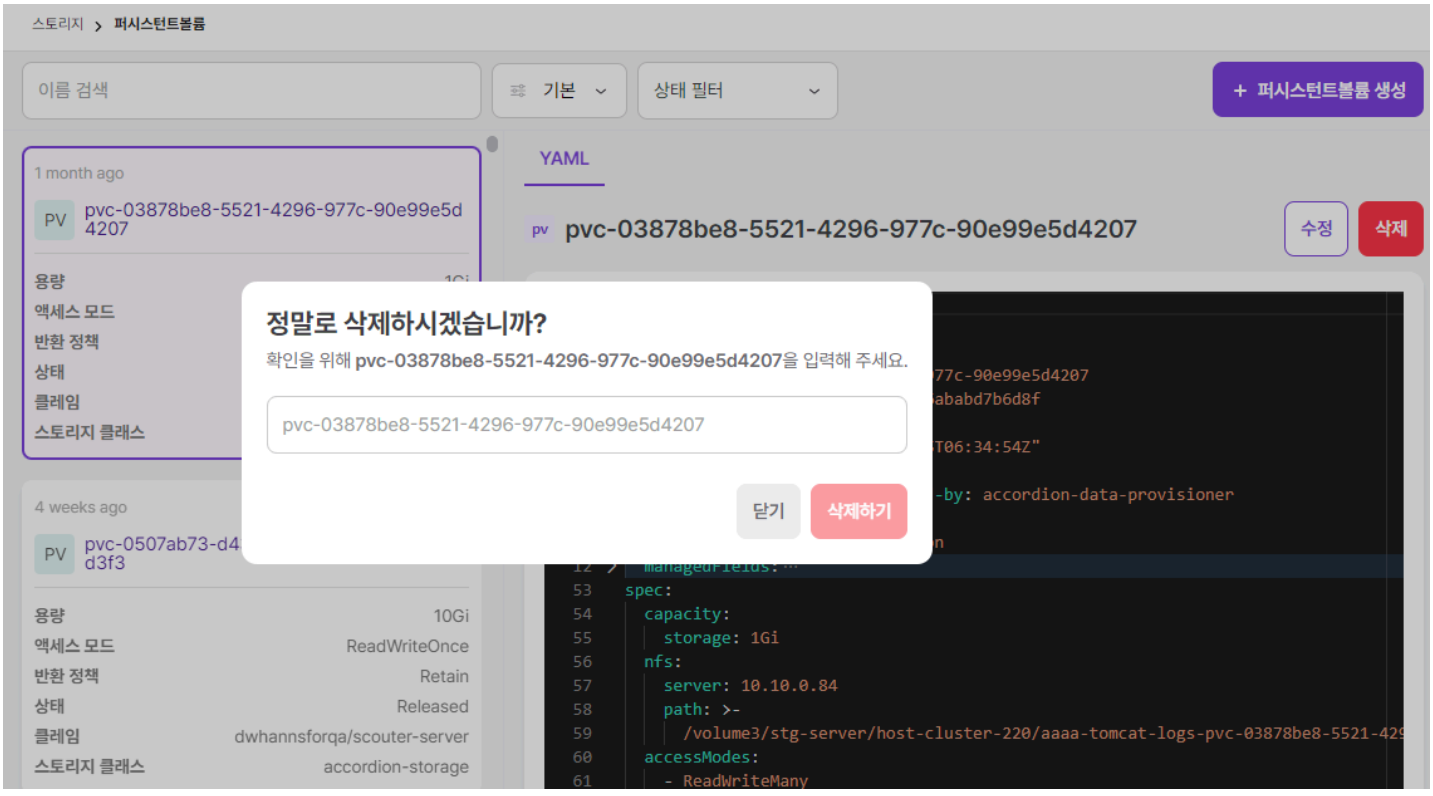



### 퍼시스턴트볼륨 수정

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

### 퍼시스턴트볼륨 삭제

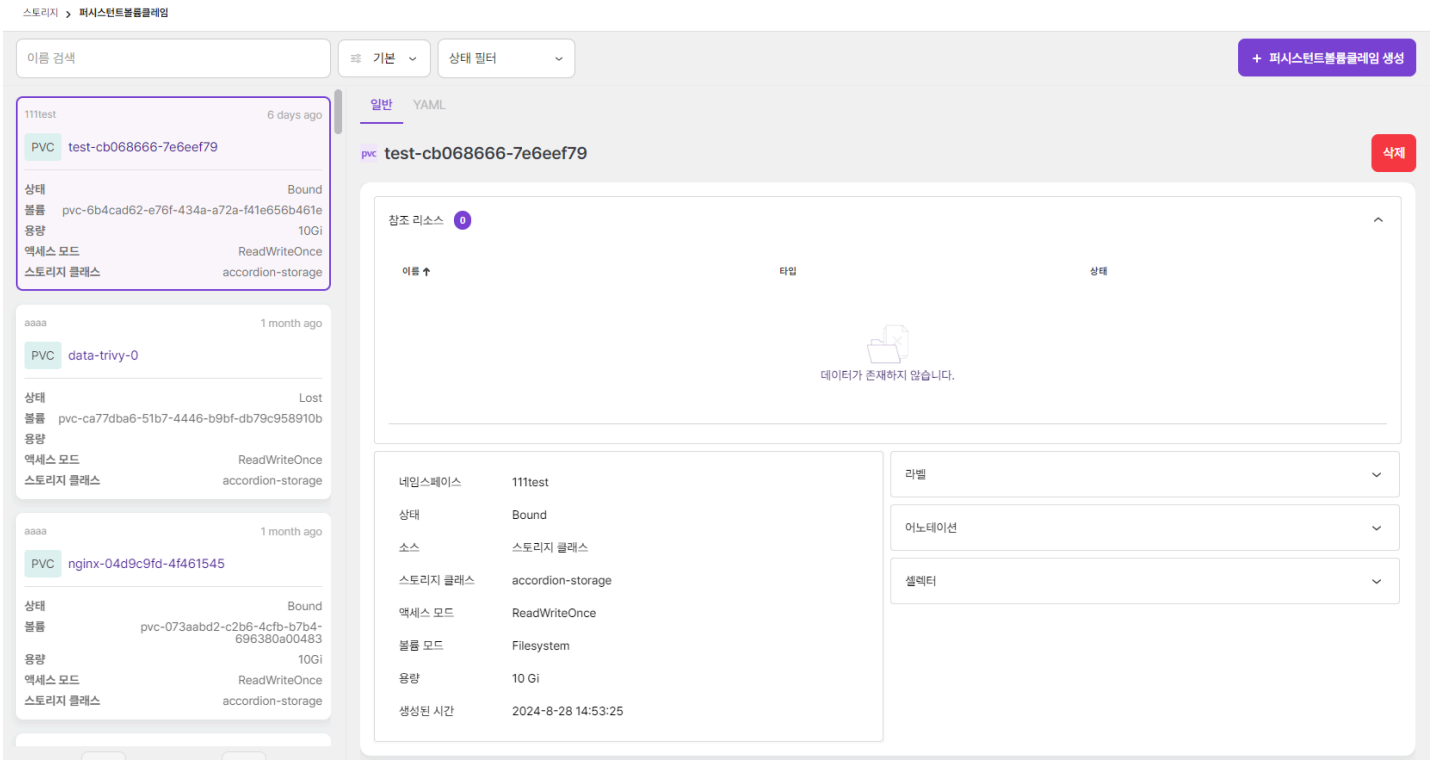
삭제하려는 퍼시스턴트볼륨을 선택하고 우측의 삭제 버튼을 선택한다.




모달에서 퍼시스턴트볼륨 이름을 입력하여 삭제한다.

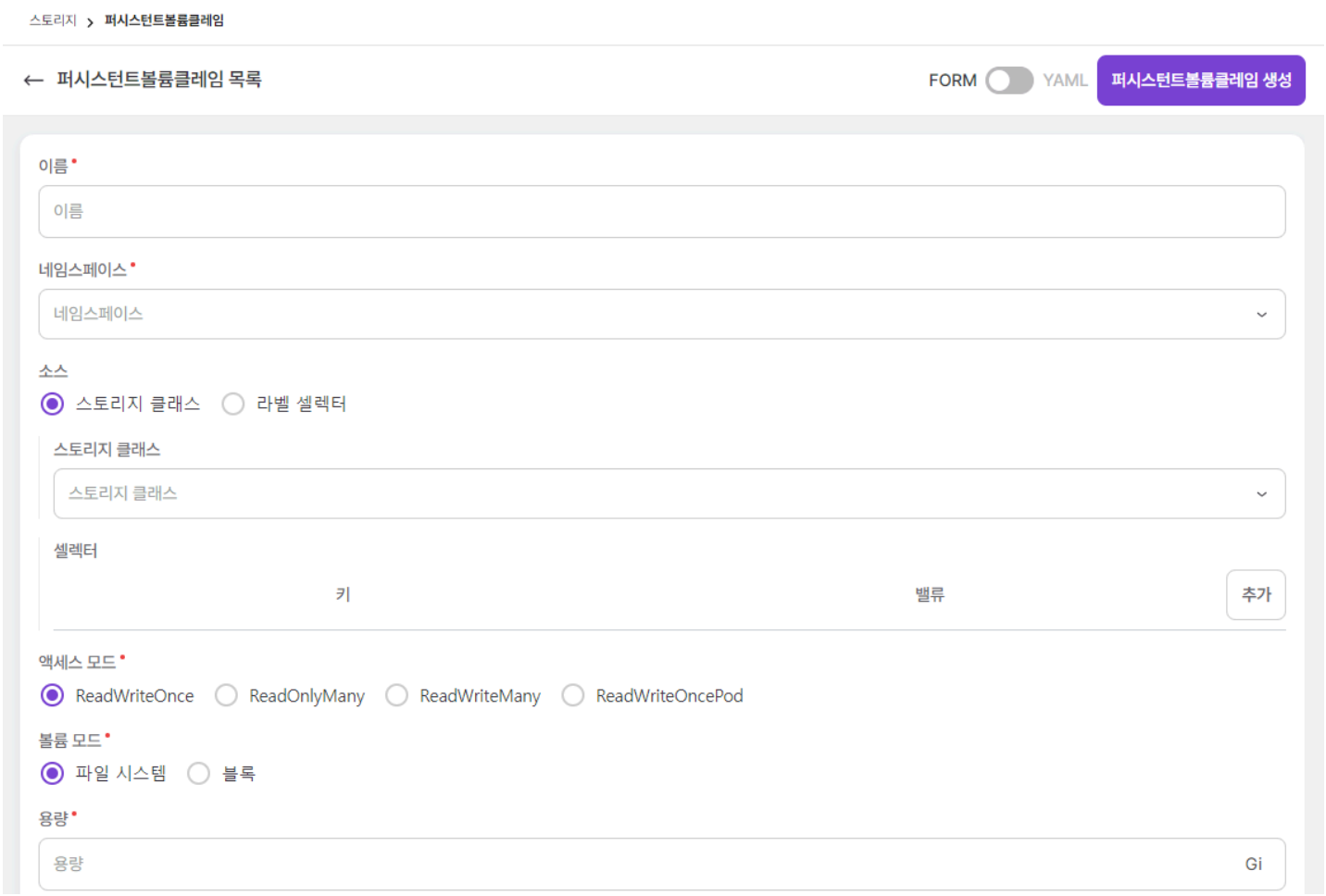
## 4.2.9.2. 퍼시스턴트볼륨클레임

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




### 퍼시스턴트볼륨클레임 생성

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

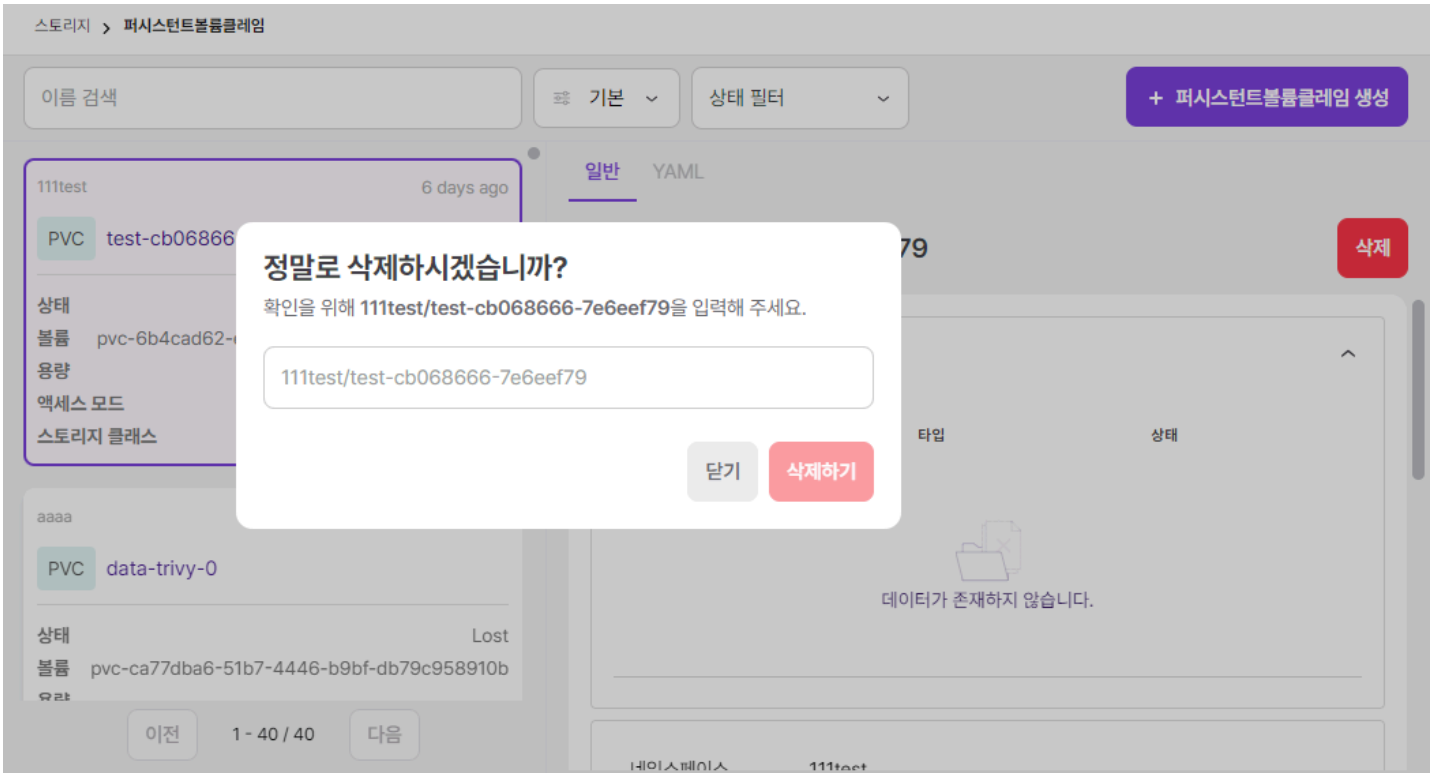

스토리지 › 퍼시스턴트볼륨클레임
← 퍼시스턴트볼륨클레임 목록
FORM YAML
퍼시스턴트볼륨클레임 생성
이름 *
이름
네임스페이스 *
네임스페이스
소스
스토리지 클래스
라벨 셀렉터
스토리지 클래스
스토리지 클래스
셀렉터
키
밸류
추가
액세스 모드 *
ReadWriteOnce
ReadOnlyMany
ReadWriteMany
ReadWriteOncePod
볼륨 모드 *
파일 시스템
블록
용량 *
용량
Gi

### 퍼시스턴트볼륨클레임 수정

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

### 퍼시스턴트볼륨클레임 삭제

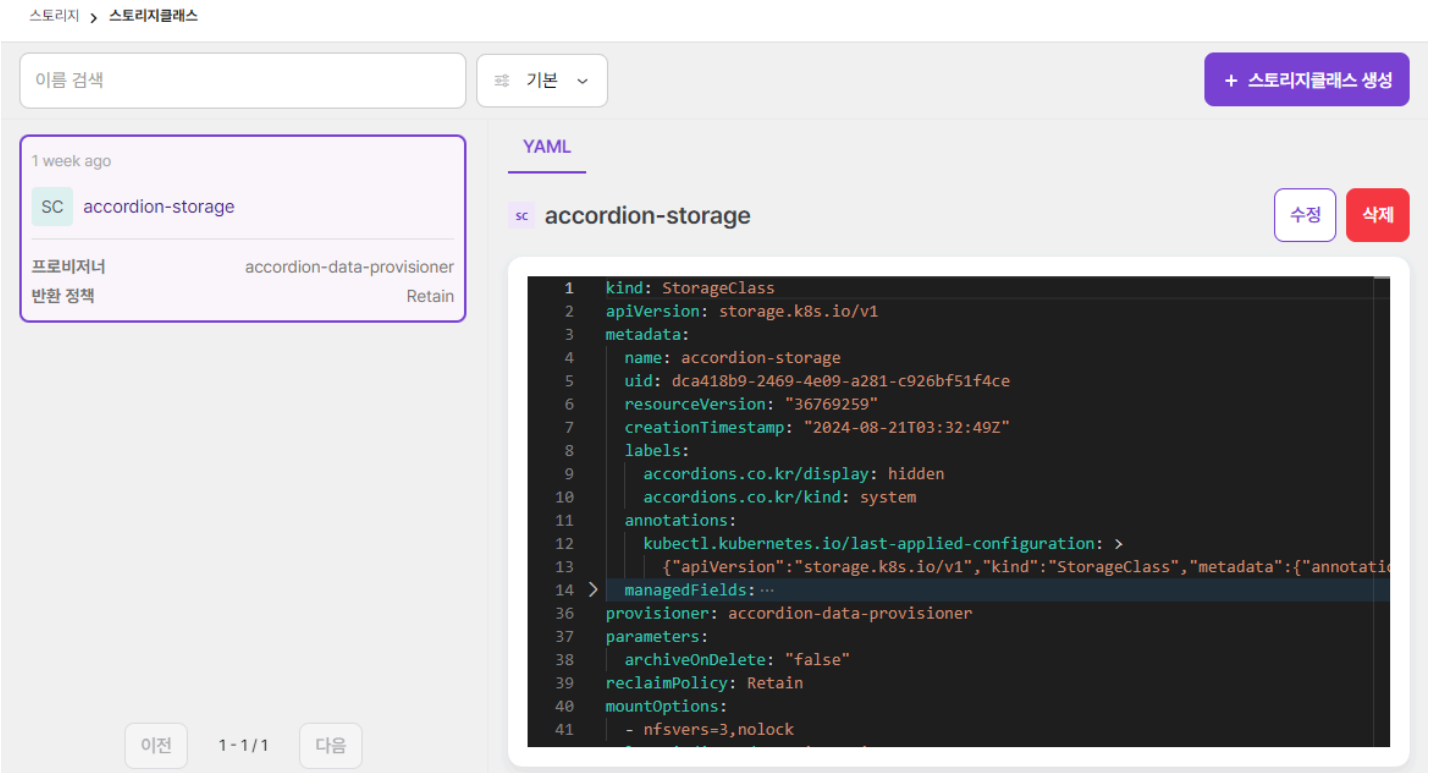
삭제하려는 퍼시스턴트볼륨클레임을 선택하고 우측의 `삭제` 버튼을 선택한다.




모달에서 네임스페이스와 퍼시스턴트볼륨클레임 이름을 입력하여 삭제한다.

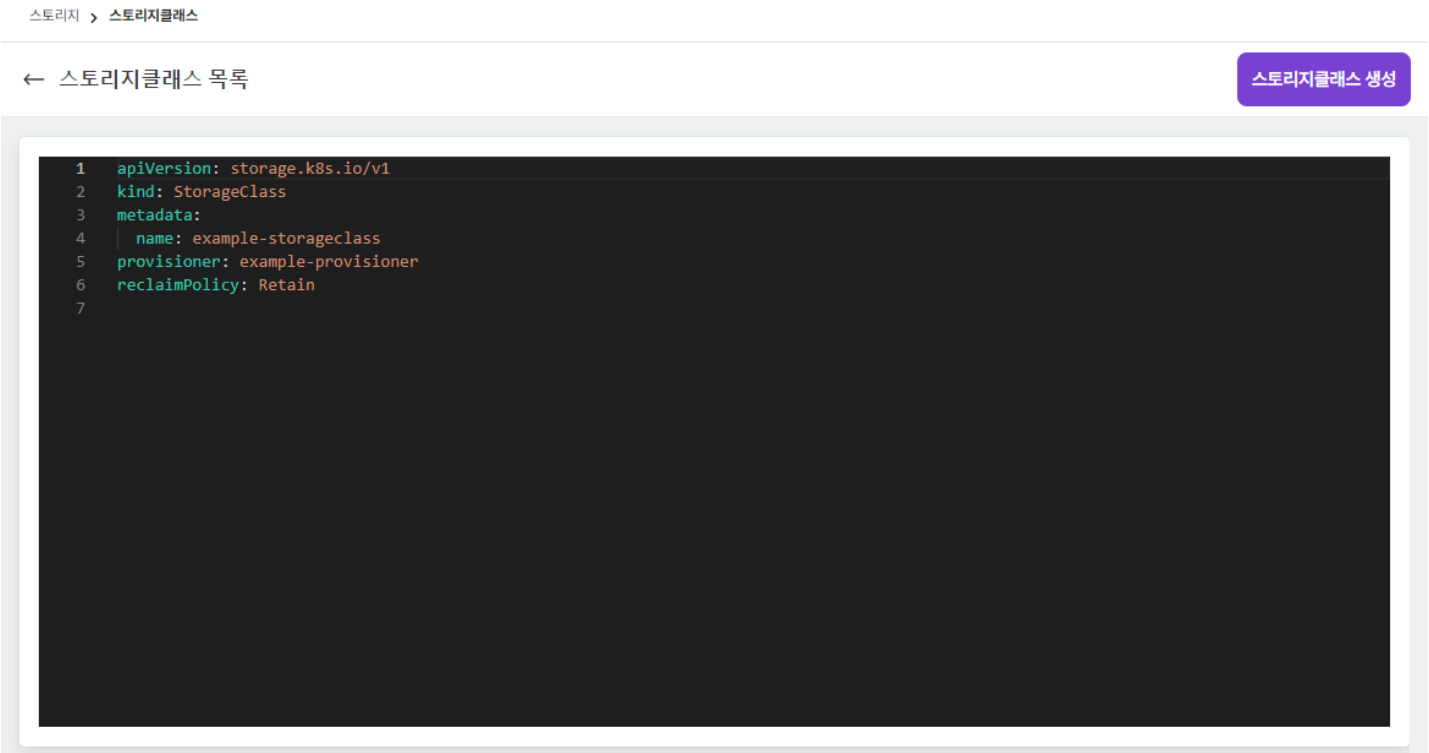
### 4.2.9.3. 스토리지클래스

스토리지클래스는 해당 스토리지클래스에 속하는 퍼시스턴트볼륨을 동적으로 프로비저닝한다.




### 스토리지클래스 생성

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

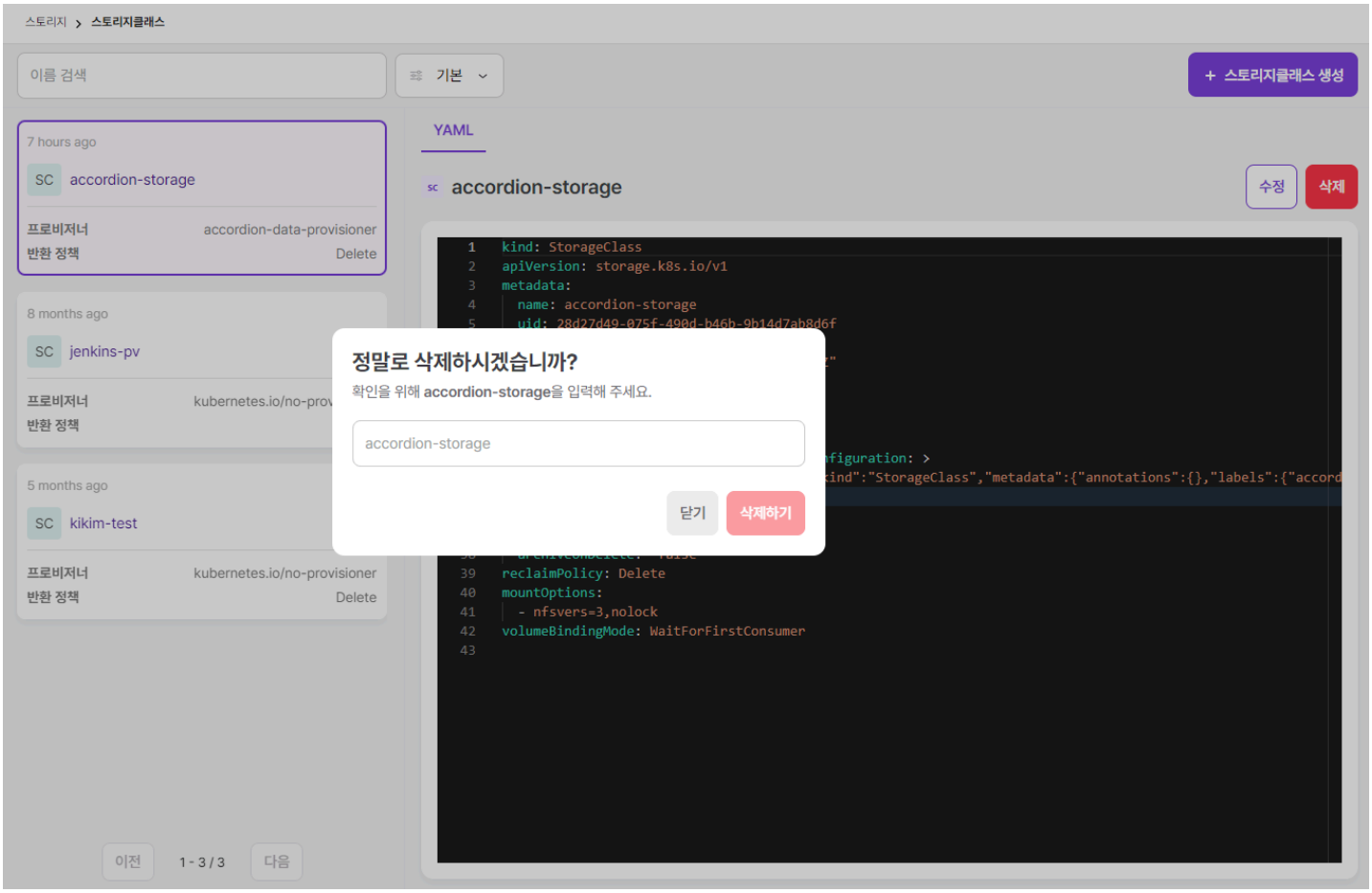


## 스토리지클래스 수정

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

## 스토리지클래스 삭제

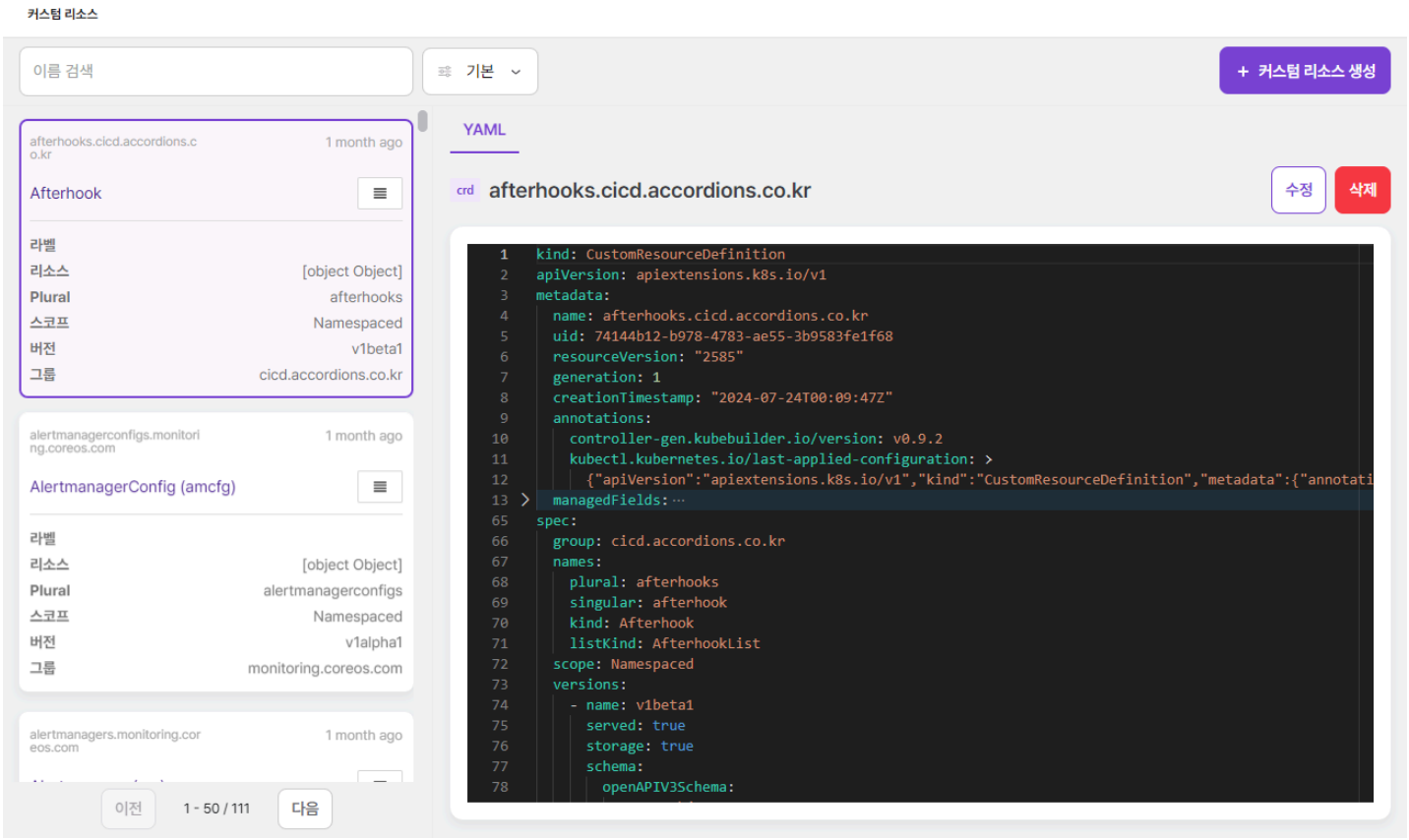
삭제하려는 스토리지클래스를 선택하고 우측의 `삭제` 버튼을 선택한다.




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

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

커스텀 리소스

← 커스텀 리소스 목록

커스텀 리소스 생성

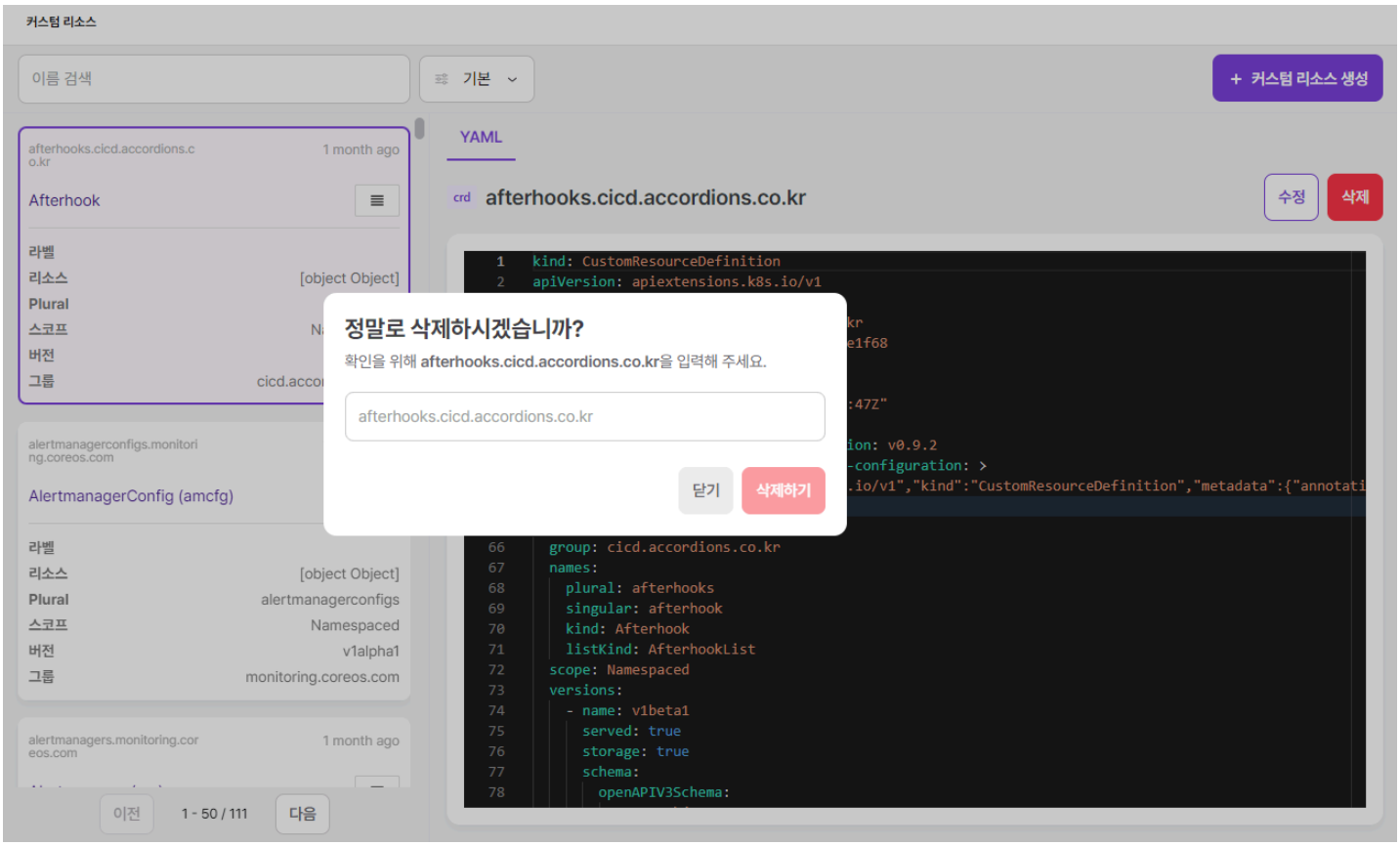
```
1  apiVersion: apiextensions.k8s.io/v1
2  kind: CustomResourceDefinition
3
```

### 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.



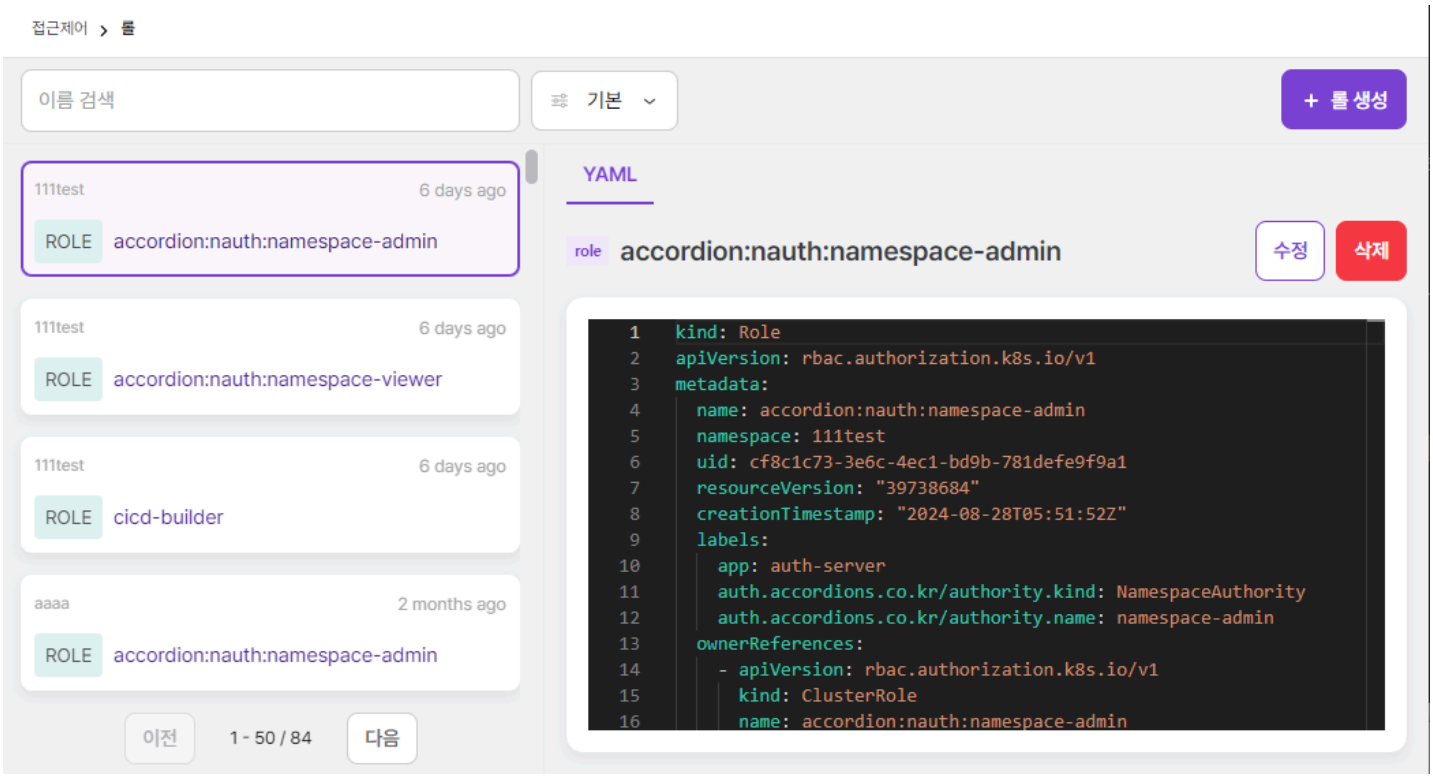

모달에서 네임스페이스와 커스텀 리소스 이름을 입력하여 삭제한다.

## 4.2.11. 접근제어

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

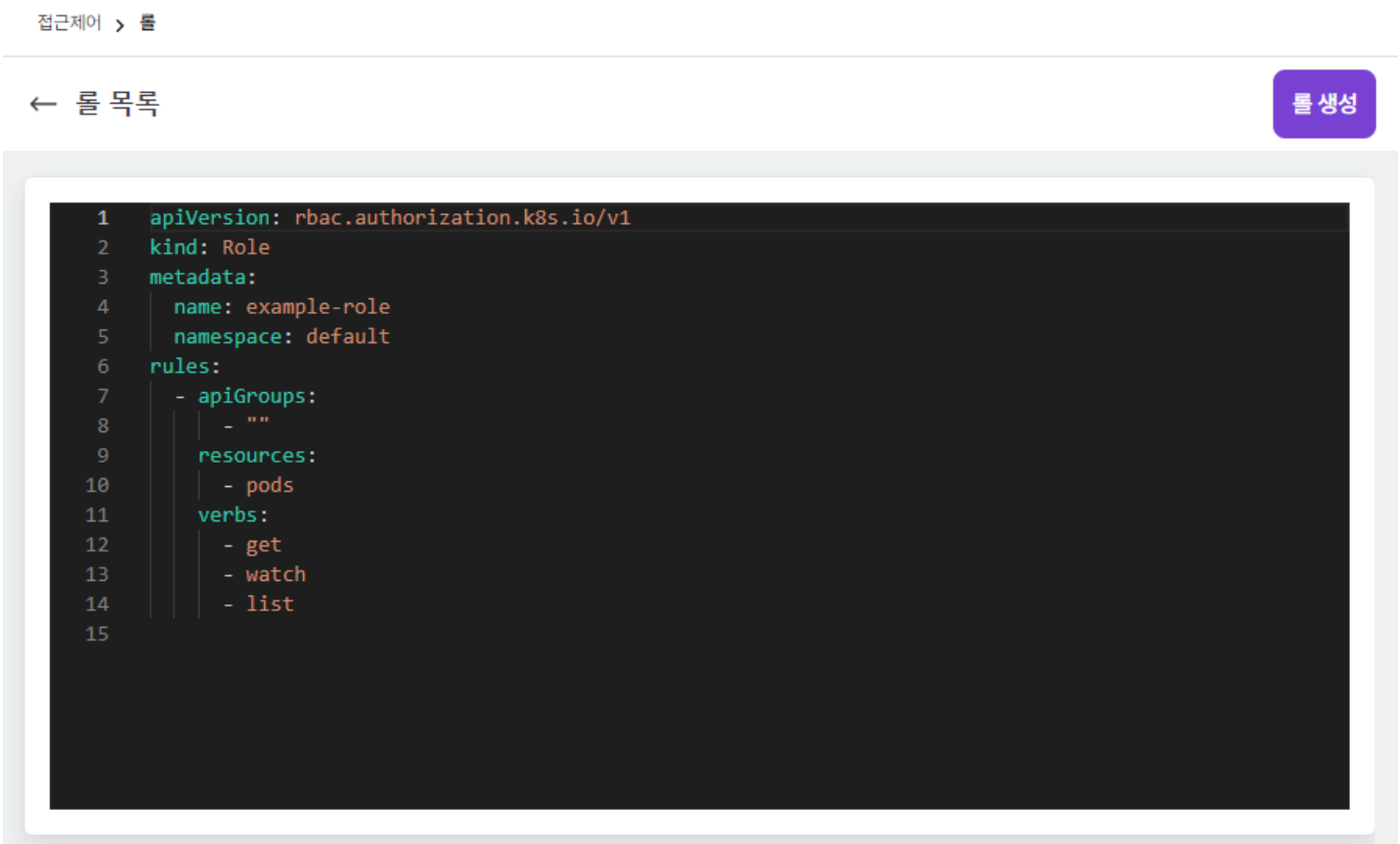
### 4.2.11.1. 롤

롤은 특정 네임스페이스의 API나 리소스에 대한 권한을 명시한다.




#### 롤 생성

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

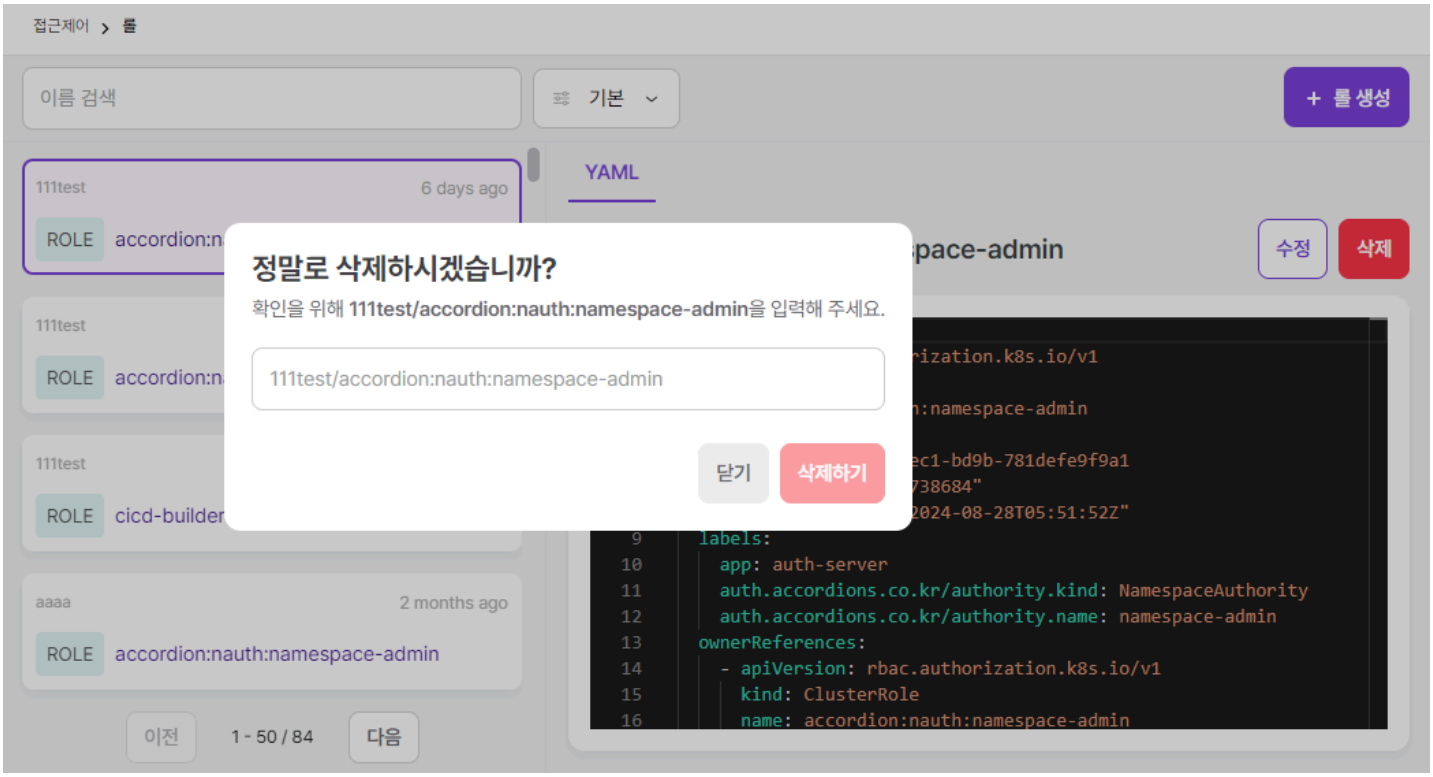


### 롤 수정

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

### 롤 삭제

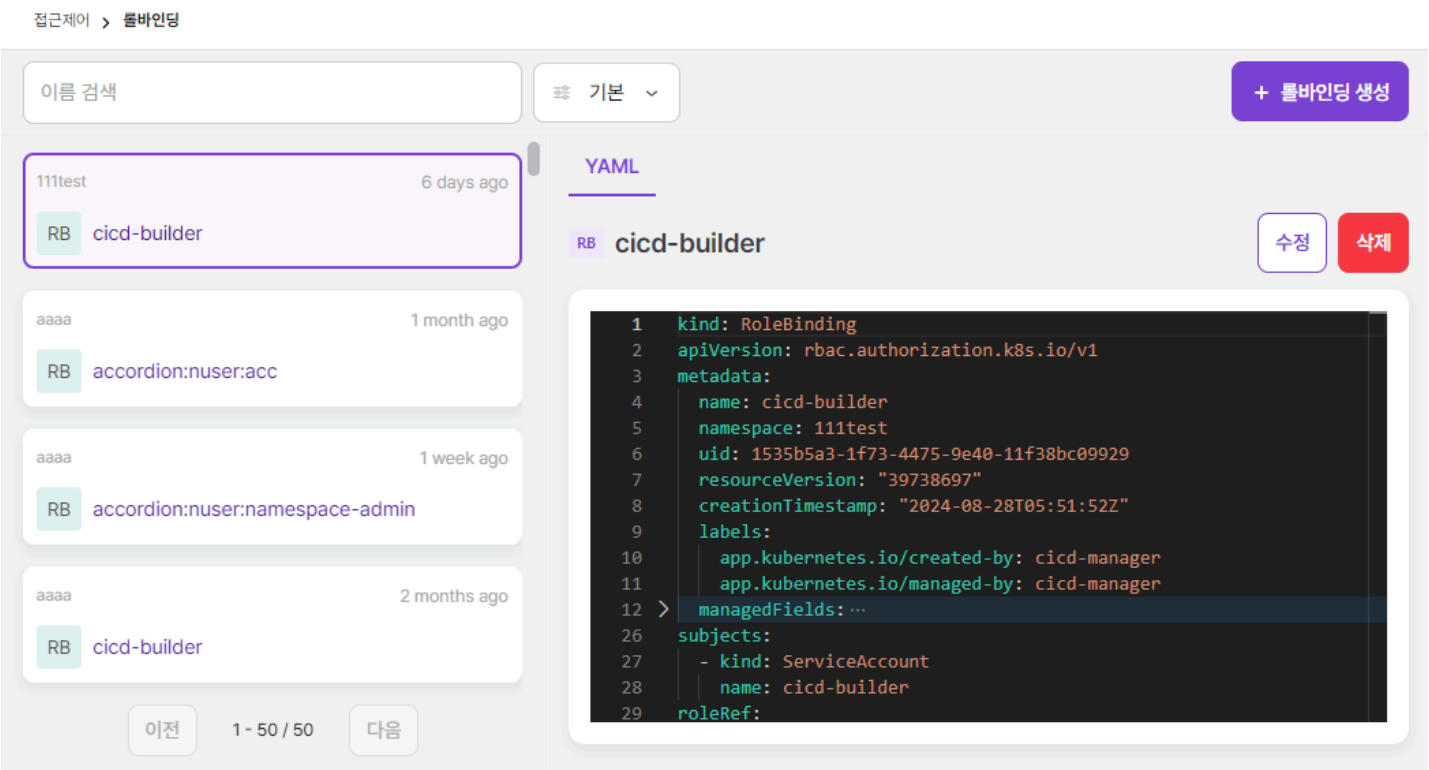
삭제하려는 롤을 선택하고 우측의 `삭제` 버튼을 선택한다.




모달에서 네임스페이스와 롤 이름을 입력하여 삭제한다.

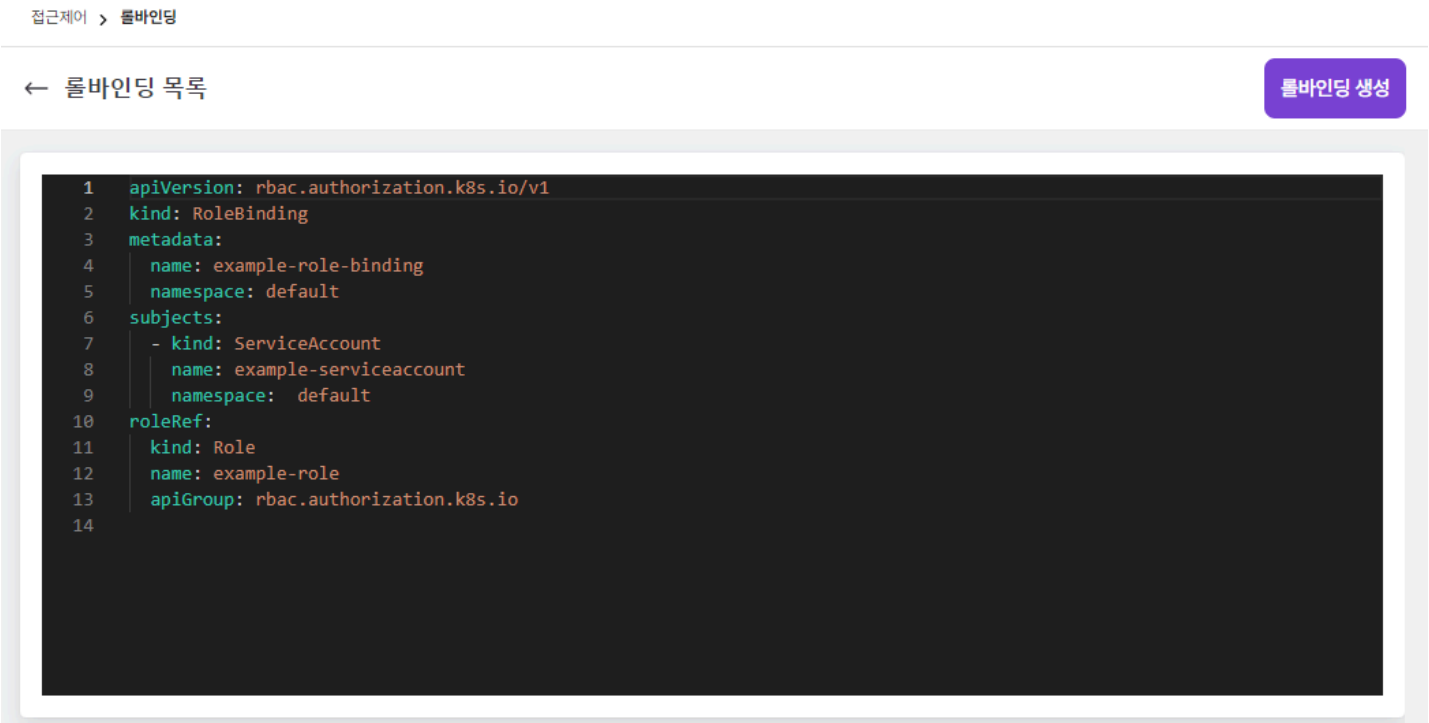
## 4.2.11.2. 롤바인딩

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




### 롤바인딩 생성

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

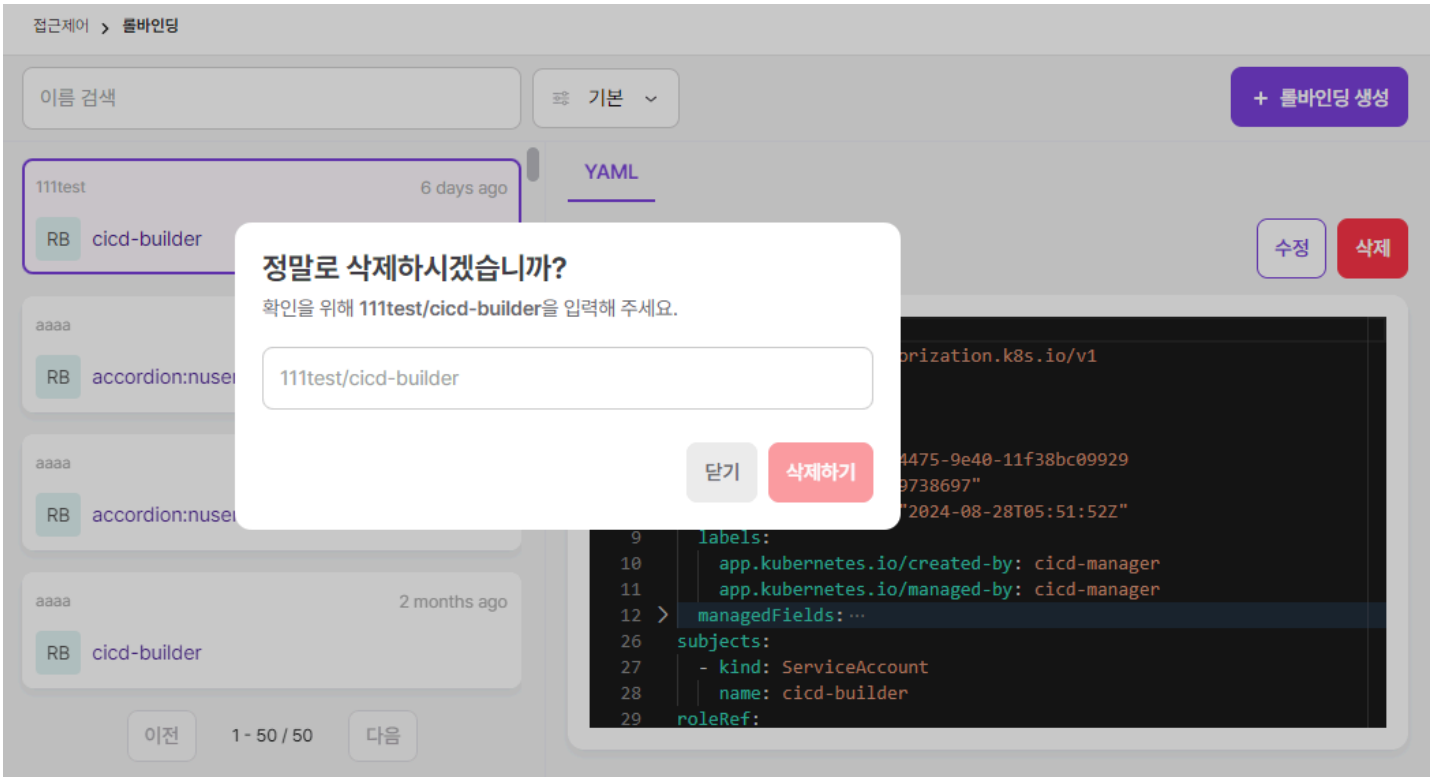


### 롤바인딩 수정

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

### 롤바인딩 삭제

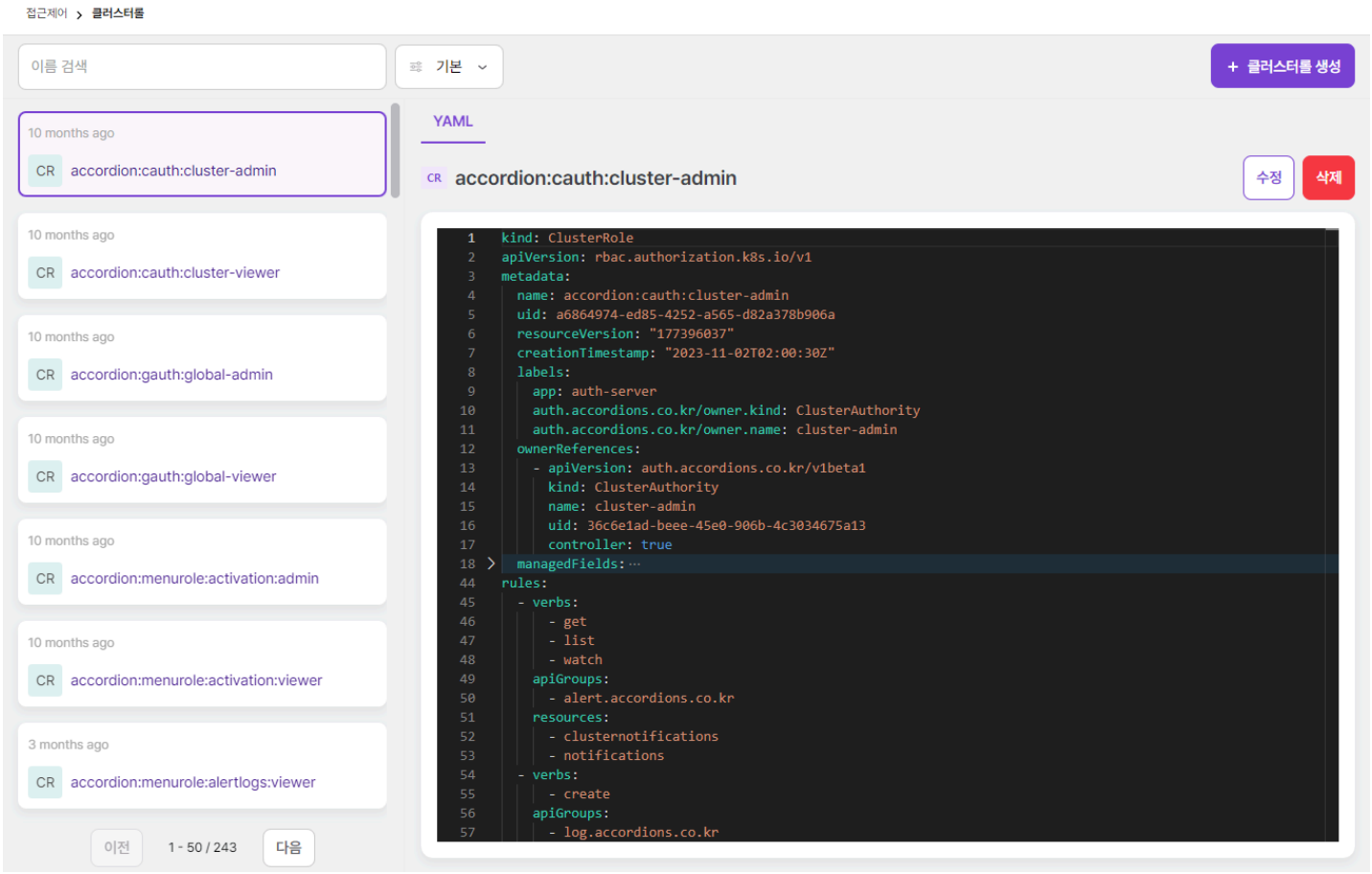
삭제하려는 롤바인딩을 선택하고 우측의 `삭제` 버튼을 선택한다.




모달에서 네임스페이스와 롤바인딩 이름을 입력하여 삭제한다.

## 4.2.11.3. 클러스터롤

클러스터롤은 클러스터의 API나 리소스에 대한 권한을 명시한다.




### 클러스터롤 생성

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

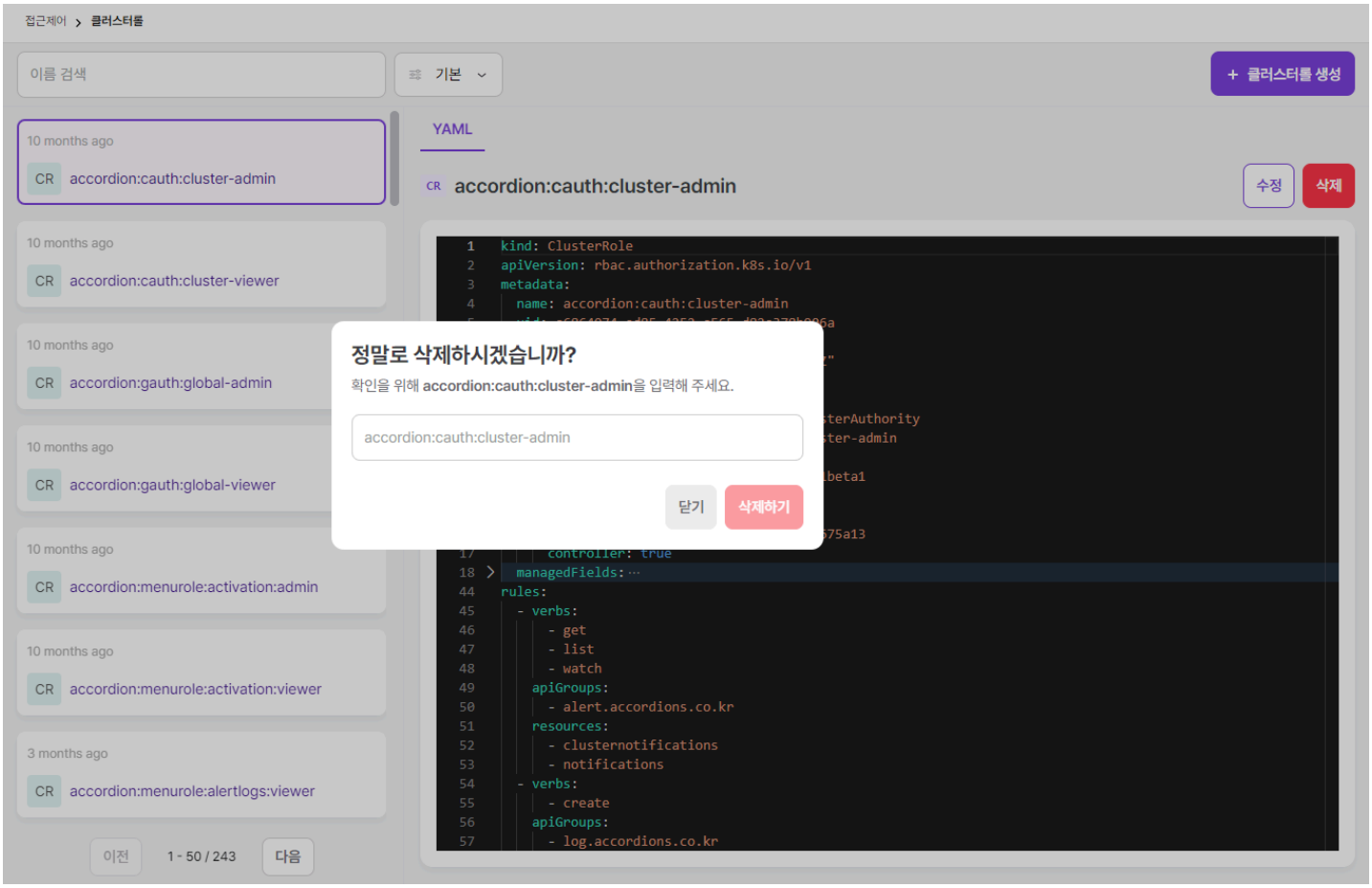
접근제어 › 클러스터롤
← 클러스터롤 목록
클러스터롤 생성
1 apiVersion: rbac.authorization.k8s.io/v1
2 kind: ClusterRole
3 metadata:
4 name: ''

## 클러스터롤 수정

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

## 클러스터롤 삭제

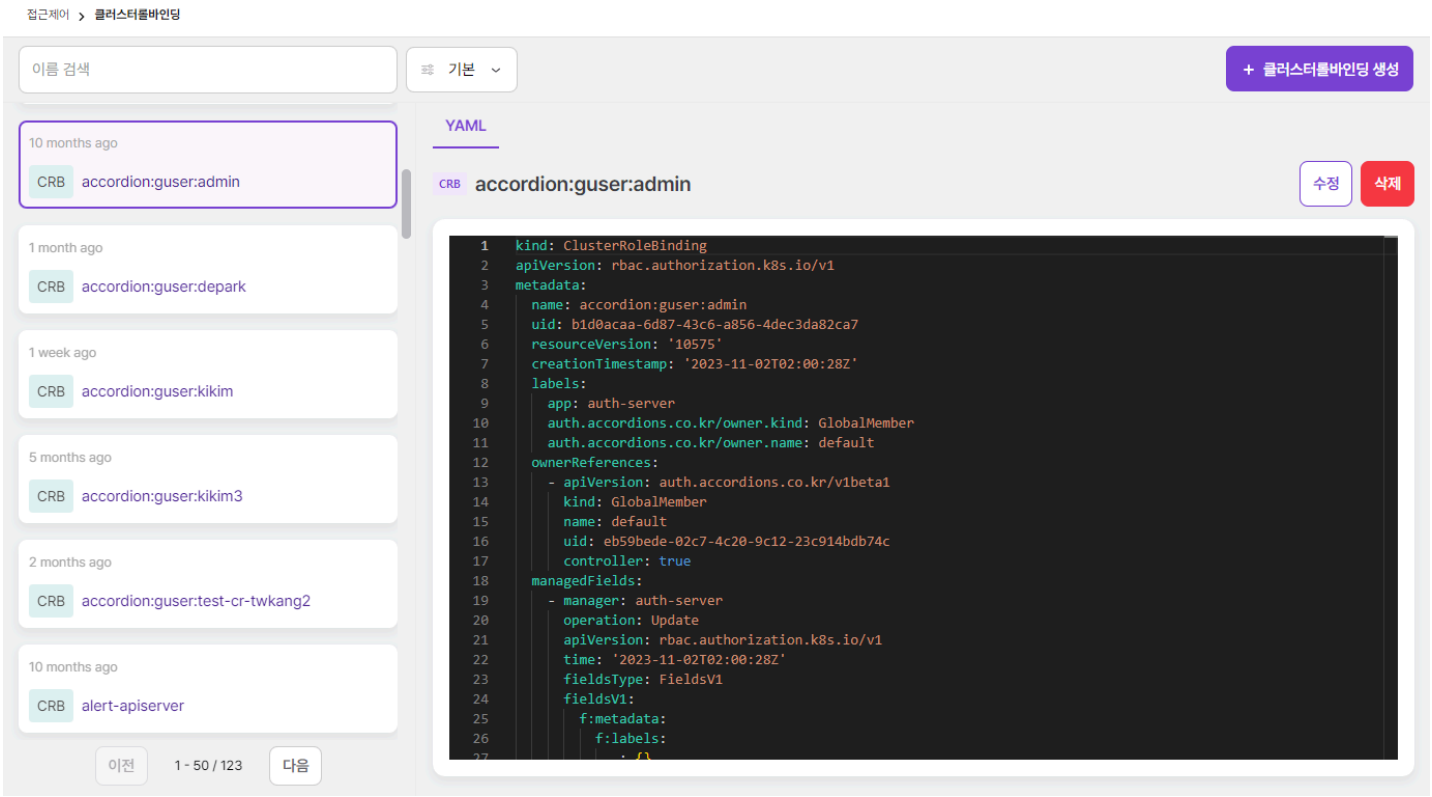
삭제하려는 클러스터롤을 선택하고 우측의 `삭제` 버튼을 선택한다.




모달에서 클러스터롤 이름을 입력하여 삭제한다.

## 4.2.11.4. 클러스터롤바인딩

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




### 클러스터롤바인딩 생성

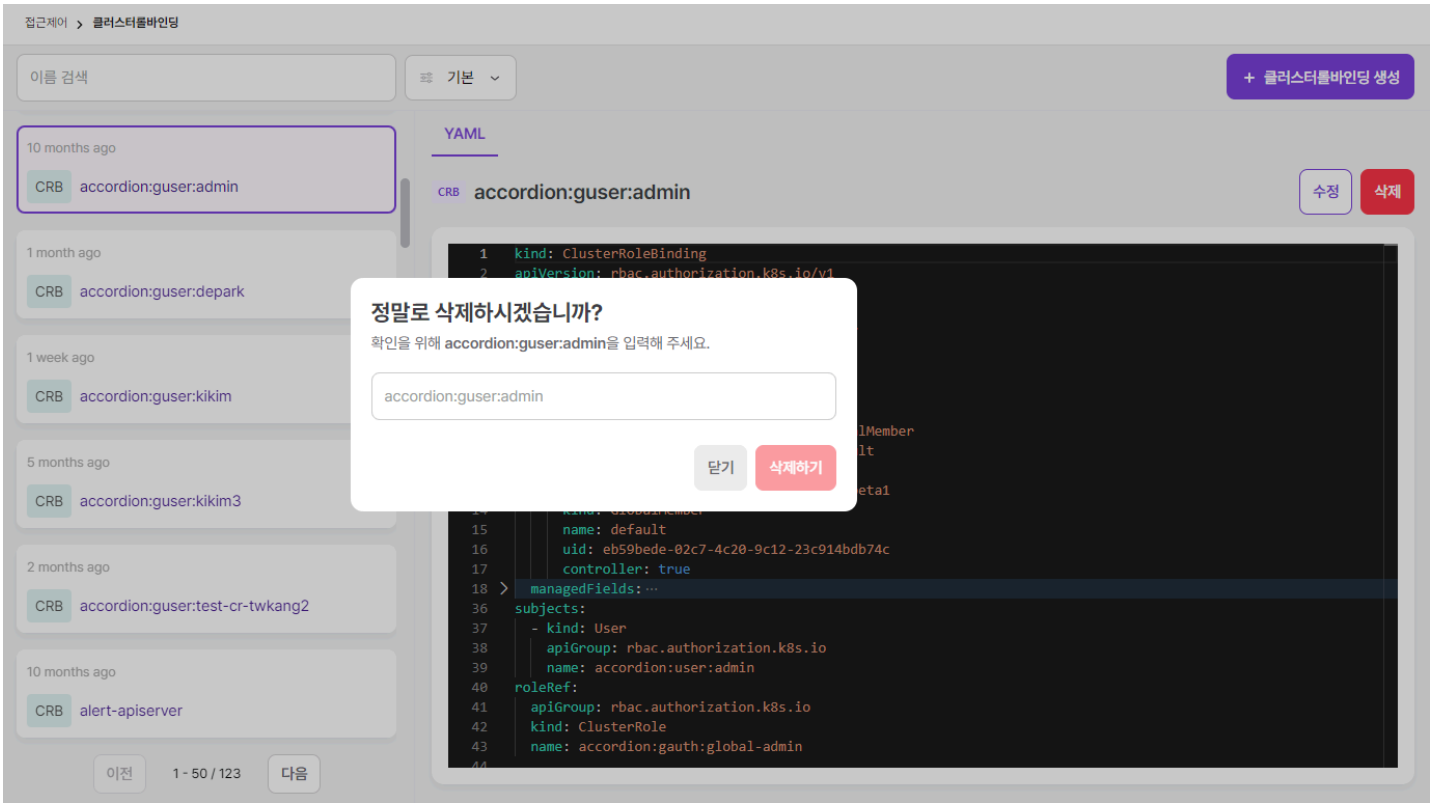
+ 클러스터롤바인딩 생성 을 선택하면 나타나는 화면에서 쿠버네티스 클러스터롤바인딩 리소스 정보를 입력하여 생성할 수 있다.

### 클러스터롤바인딩 수정

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

### 클러스터롤바인딩 삭제

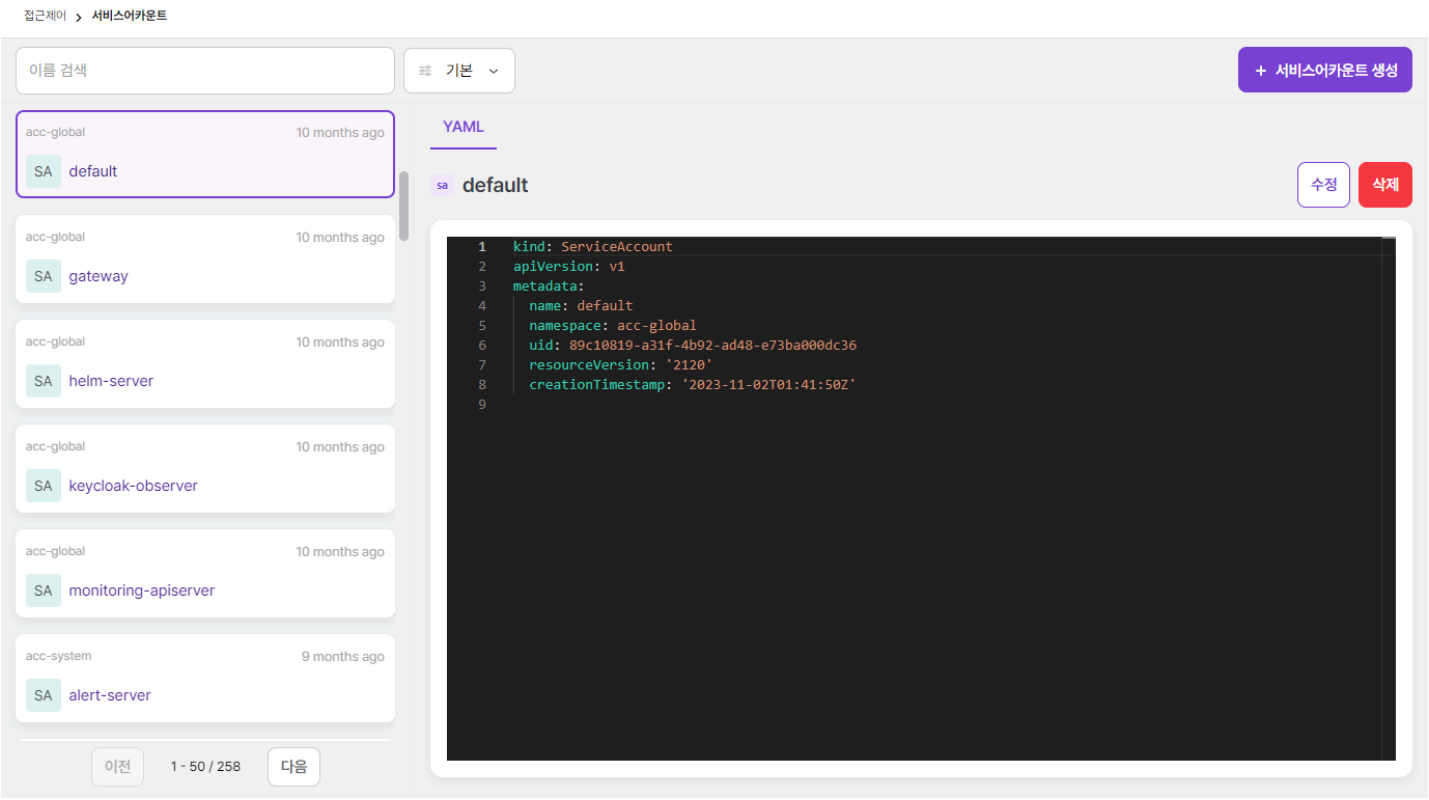
삭제하려는 클러스터롤바인딩을 선택하고 우측의 삭제 버튼을 선택한다.




모달에서 클러스터롤바인딩 이름을 입력하여 삭제한다.

## 4.2.11.5. 서비스어카운트

서비스어카운트는 쿠버네티스 API 접근 시 파드의 권한을 식별하기 위한 리소스이다.




### 서비스어카운트 생성

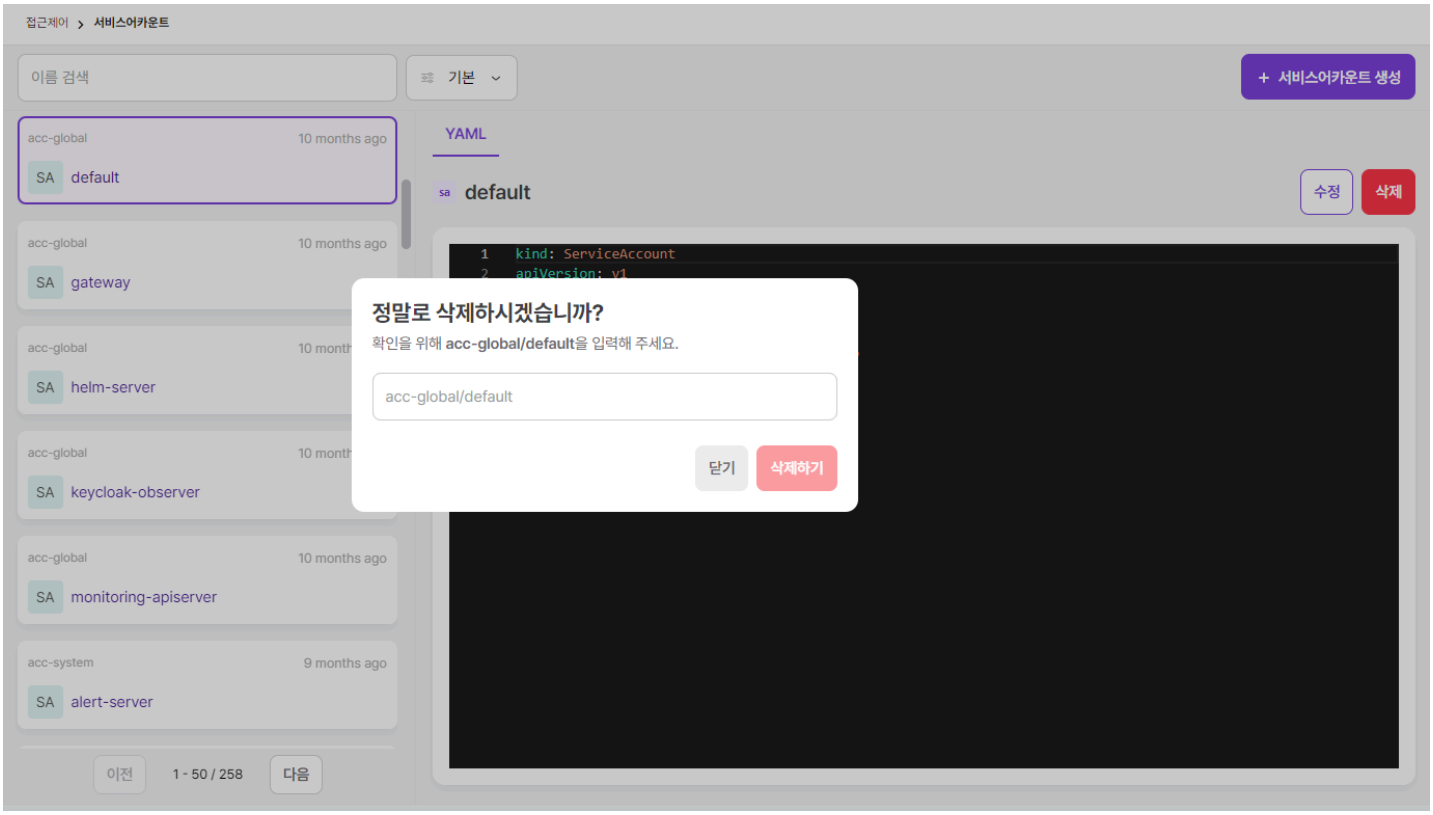
+ 서비스어카운트 생성 을 선택하면 나타나는 화면에서 쿠버네티스 서비스어카운트 리소스 정보를 입력하여 생성할 수 있다.

### 서비스어카운트 수정

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

### 서비스어카운트 삭제

삭제하려는 서비스어카운트를 선택하고 우측의 `삭제` 버튼을 선택한다.



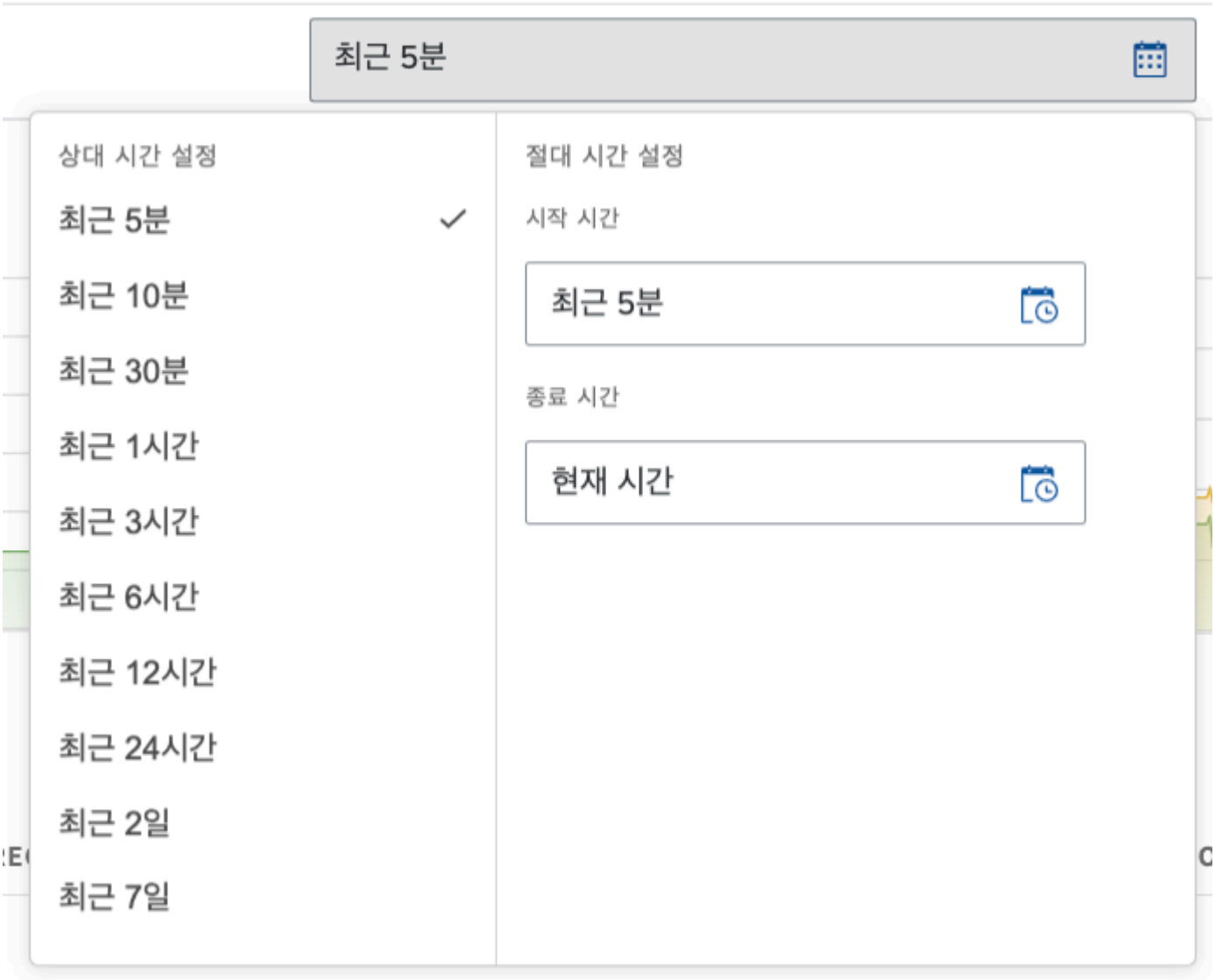

모달에서 네임스페이스와 서비스어카운트 이름을 입력하여 삭제한다.

## 4.2.12. 모니터링

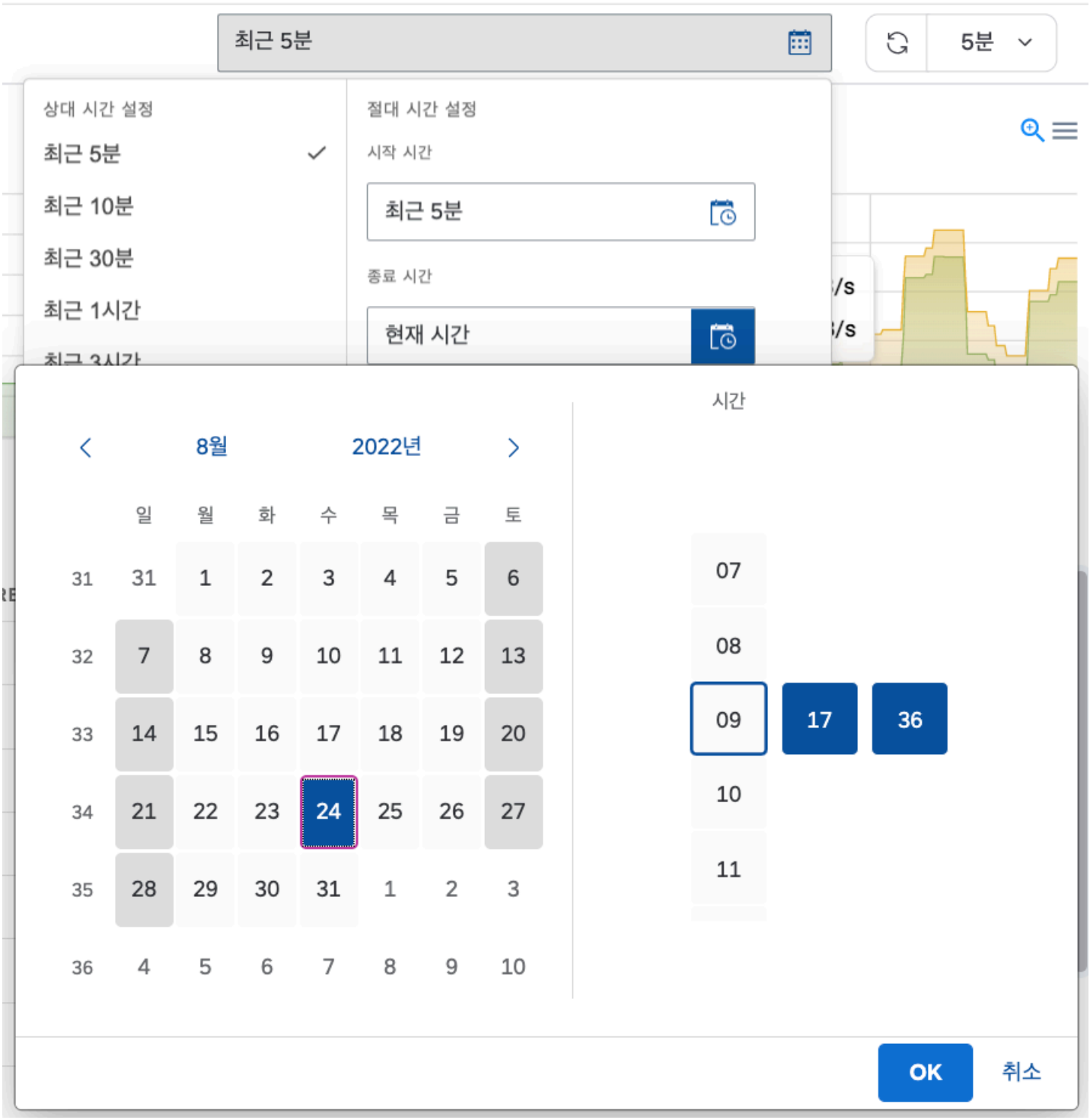
모니터링은 클러스터 별로 모니터링을 제공한다. 모니터링은 시스템 정보와 이벤트 로그, 감사 로그, 컨테이너 로그 등의 로그 정보 및 서비스 트래픽 정보에 대한 서비스 메시를 제공하여 운영자가 Accordion 내의 클러스터와 애플리케이션을 다양한 관점에서 모니터링하여 안정적인 시스템 운영할 수 있도록 한다.

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(5,15,30), 시간단위(1,3,6,12,24), 일단위(2,7)로 지정할 수 있다.

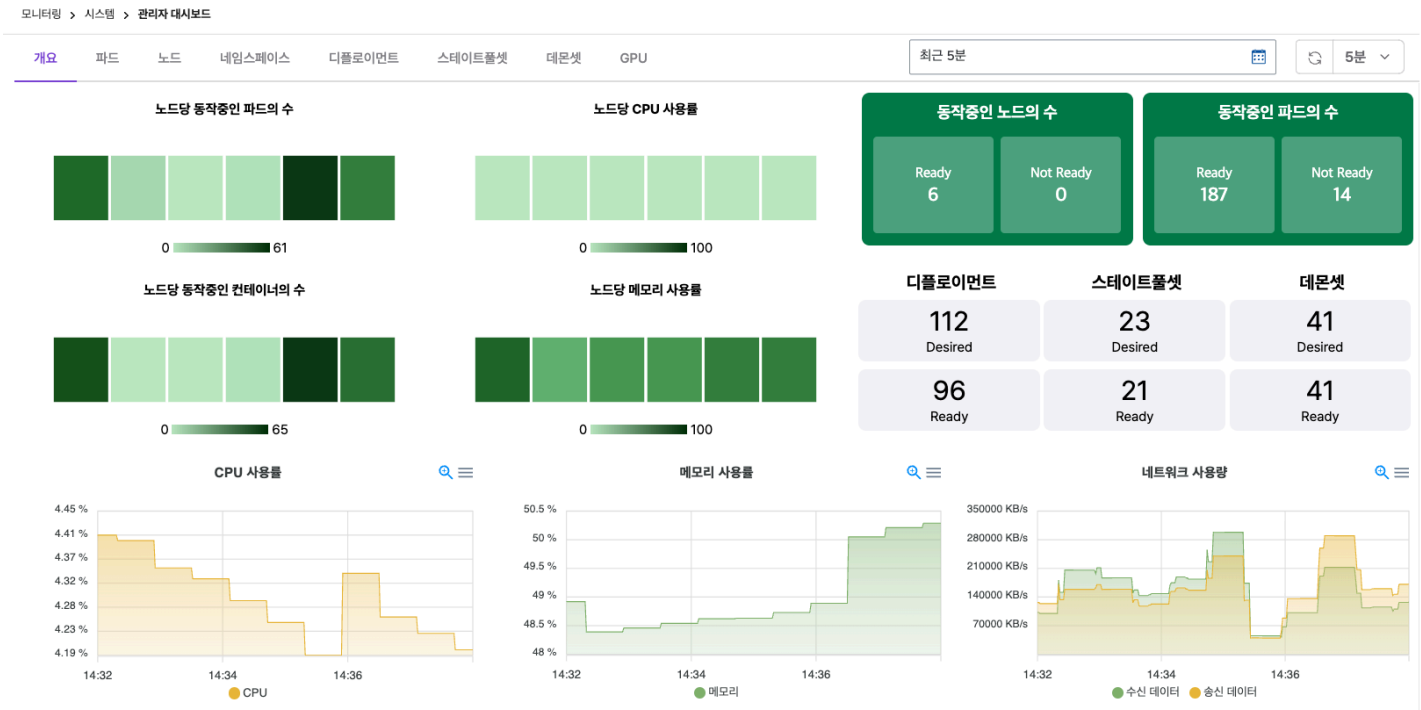


둘째, 캘린더를 선택하여 원하는 검색 날짜/시간의 정보를 확인할 수 있다. 시작 시간 캘린더에서 시작날짜를 선택하고 종료 시간 캘린더에서 종료날짜를 선택한 후 OK를 클릭하여 원하는 검색 기간을 적용한다.



## 4.2.12.1. 시스템

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

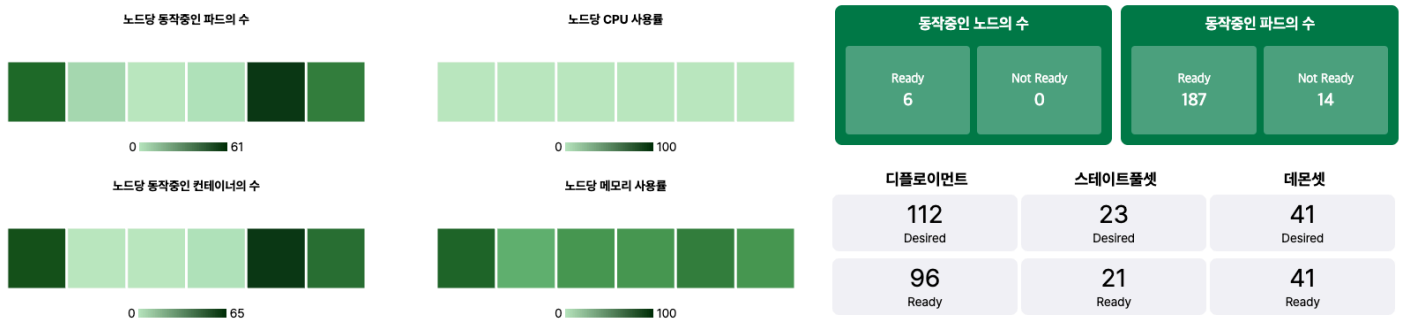



| 네임스페이스 | CPU 사용량 | CPU 요청 | CPU 제한 | 메모리 사용량(MI) | 메모리 요청(MI) | 메모리 제한(MI) | 네트워크 인(MB/S) | 네트워크 아웃(MB/S) |
|---|---|---|---|---|---|---|---|---|
| acc-global | 0.28 | 0 | - | 3422.83 | 0 | - | 6.83 | 3.07 |
| acc-system | 1.46 | 0 | - | 26289.56 | 0 | - | 25.81 | 30.86 |
| default | 0 | 0 | - | 8.11 | 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.



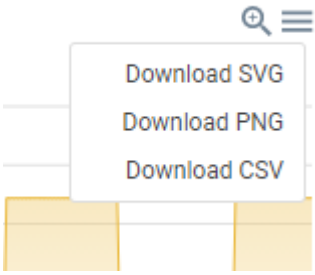

| 항목 | 설명 |
|---|---|
| 노드당 동작중인 파드의 수 | 클러스터 내 노드 내 동작중인 파드의 개수를 보여준다.<br>상대적으로 파드의 개수가 많을 수록 진한 색으로 표시된다. |
| 노드당 CPU 사용률 | 클러스터 내 노드 내 CPU 사용률을 보여준다.<br>절대적으로 사용률이 높을수록 진한 색으로 표시된다. |
| 노드당 동작중인 컨테이너의 수 | 클러스터 내 노드 내 동작중인 컨테이너의 수를 보여준다.<br>상대적으로 컨테이너의 개수가 많을 수록 진한 색으로 표시된다. |
| 노드당 메모리 사용률 | 클러스터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

| 항목 | 설명 |
| --- | --- |
| 데몬셋 | 클러스터 내 데몬셋 개수를 보여준다.<br><br>**Desired** : 정의된 데몬셋 개수를 보여준다.<br>**Ready** : 동작중인 데몬셋 개수를 보여준다. |

### 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 |

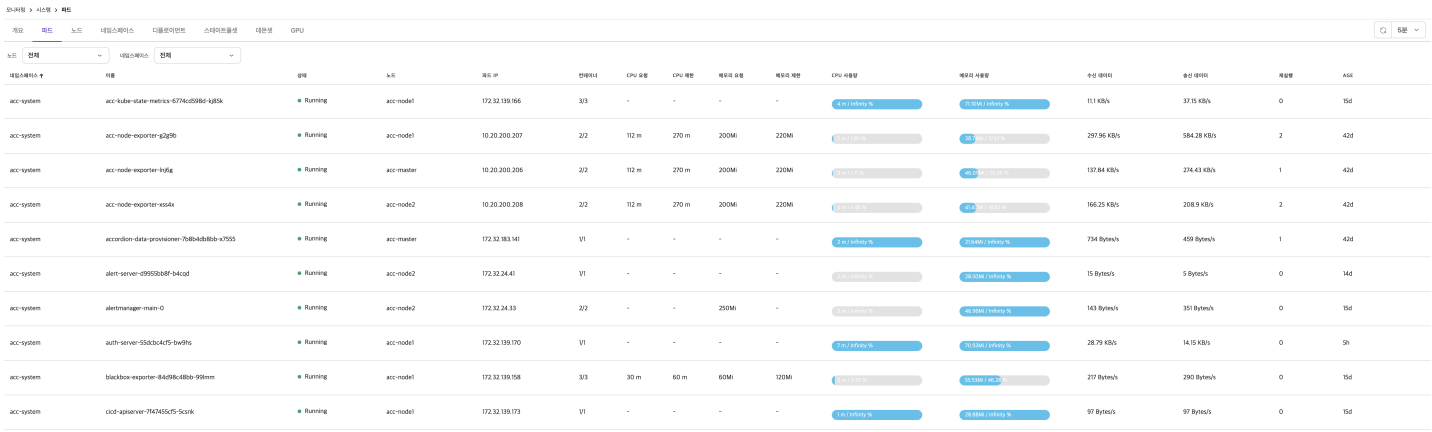
네임스페이스 정보에 대한 내용은 다음과 같다.

| 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.



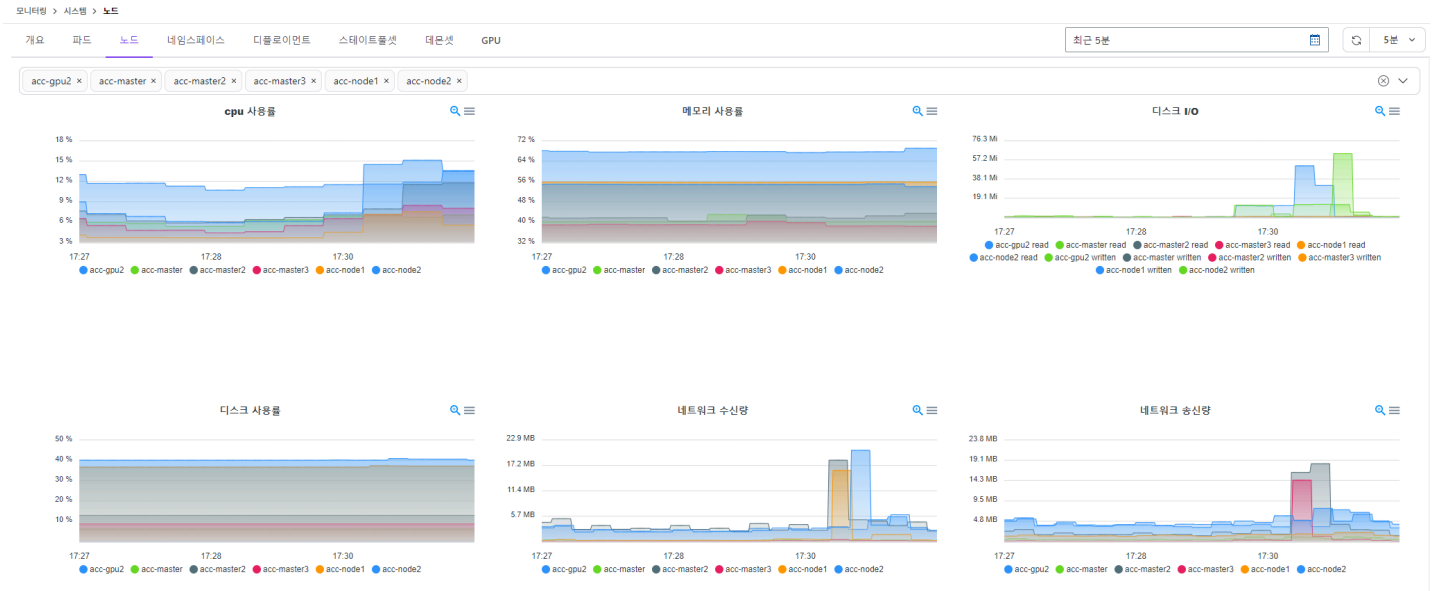
제공하는 정보는 다음과 같다.

| 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.<br><br>파드의 전체 컨테이너가 CPU 리소스 양을 지정한 경우 표시된다. 그외의 경우 "-"로 표시된다. |
| 메모리 요청 / 제한 | 파드의 Memory 리소스 Requests/Limits 할당량이다.<br><br>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.<br><br>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.<br><br>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.

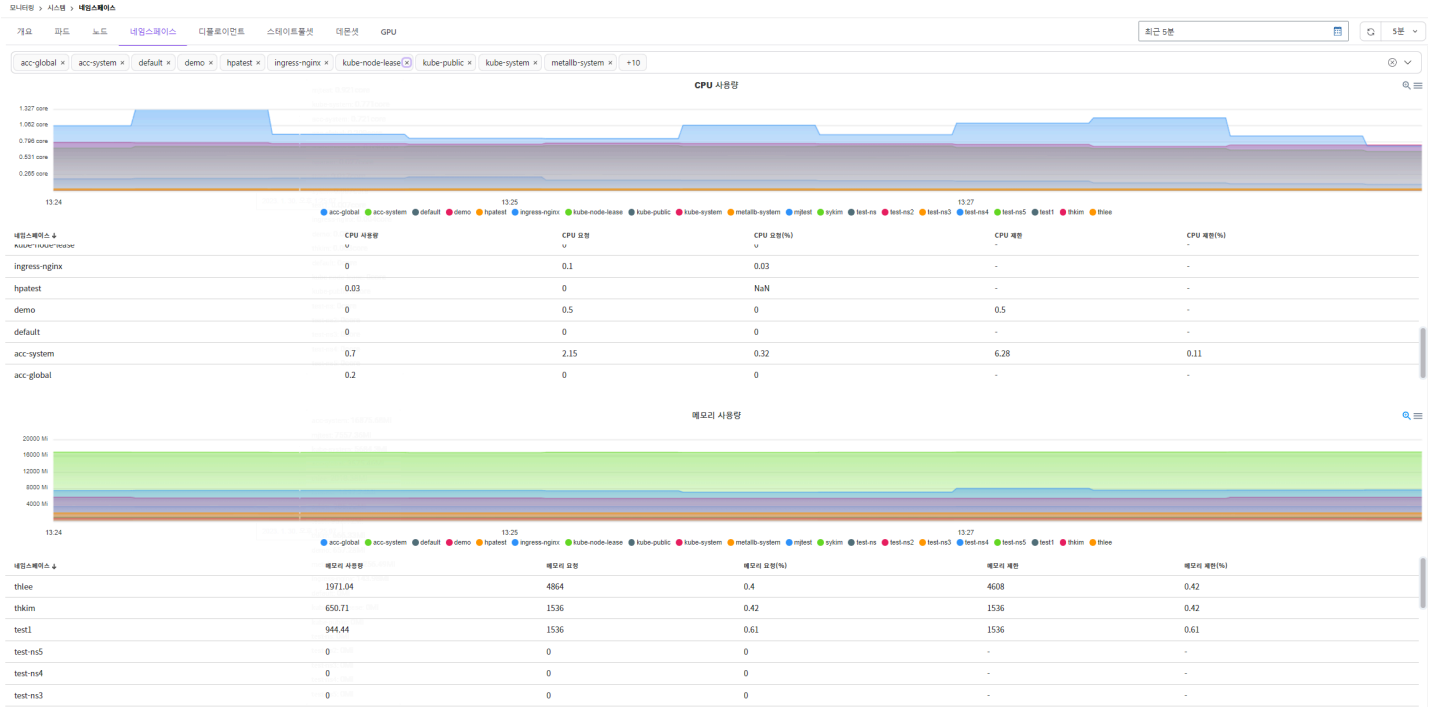



Table 8. CPU

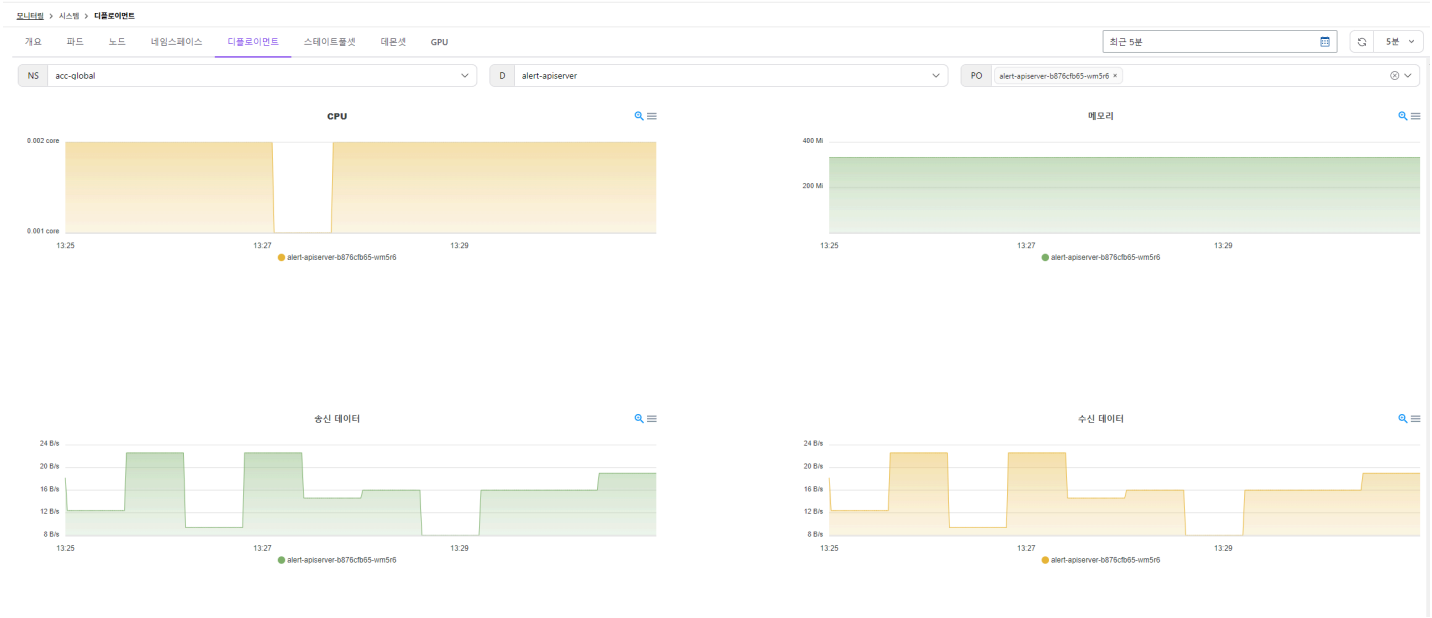
| 항목 | 설명 |
| --- | --- |
| 네임스페이스 | 네임스페이스 이름 |
| CPU 사용량 | 네임스페이스에 배포된 파드 CPU 사용량 합계 |
| CPU 요청 | 네임스페이스에 배포된 파드 CPU Requests 합계 |
| CPU 요청(%) | CPU 사용량 / CPU 요청 |
| CPU 제한 | 네임스페이스에 배포된 파드 CPU Limits 합계 |
| CPU 제한(%) | CPU 사용량 / CPU 제한 |

Table 9. 메모리

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

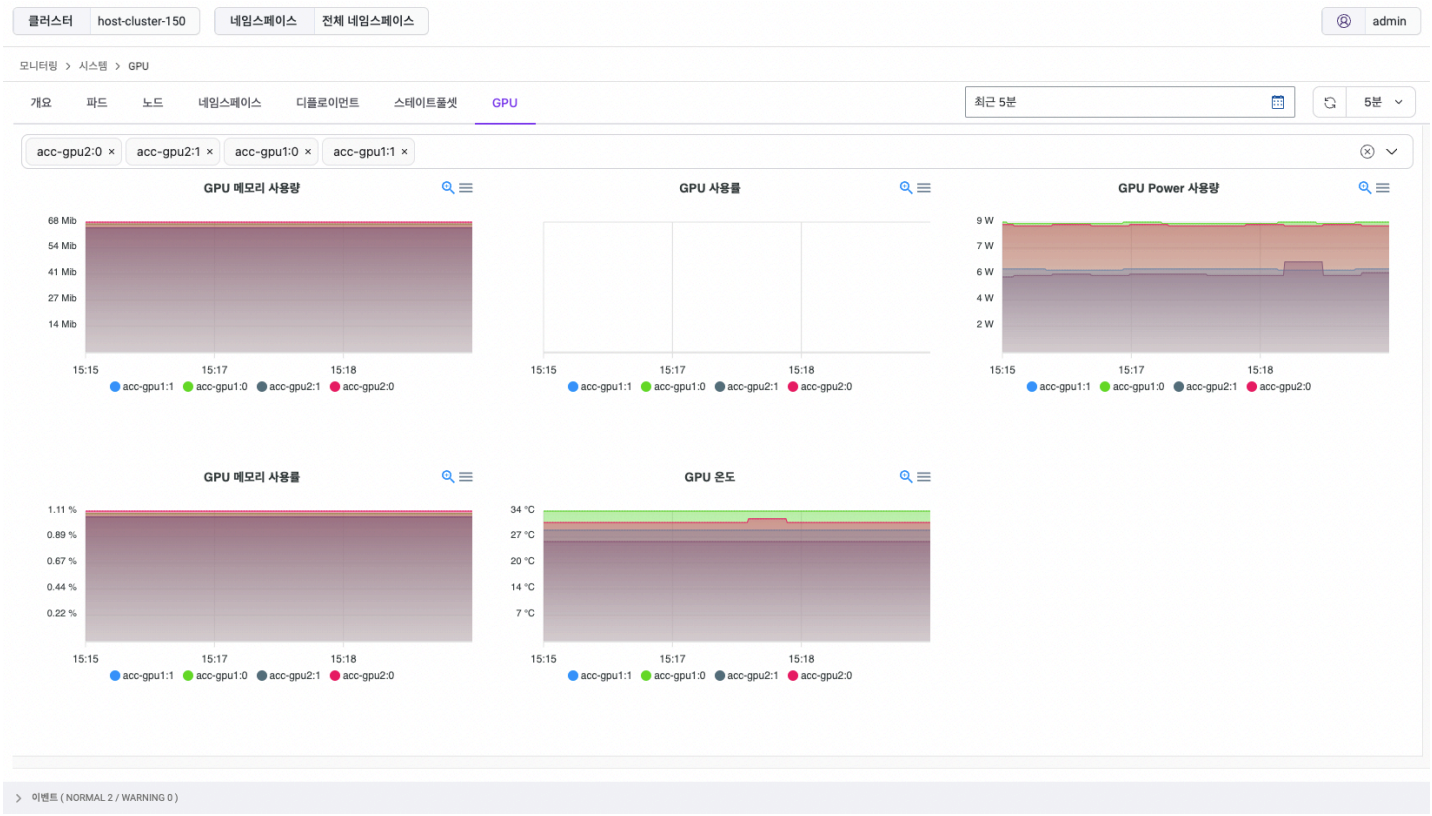
## 디플로이먼트 / 스테이트풀셋 / 데몬셋

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



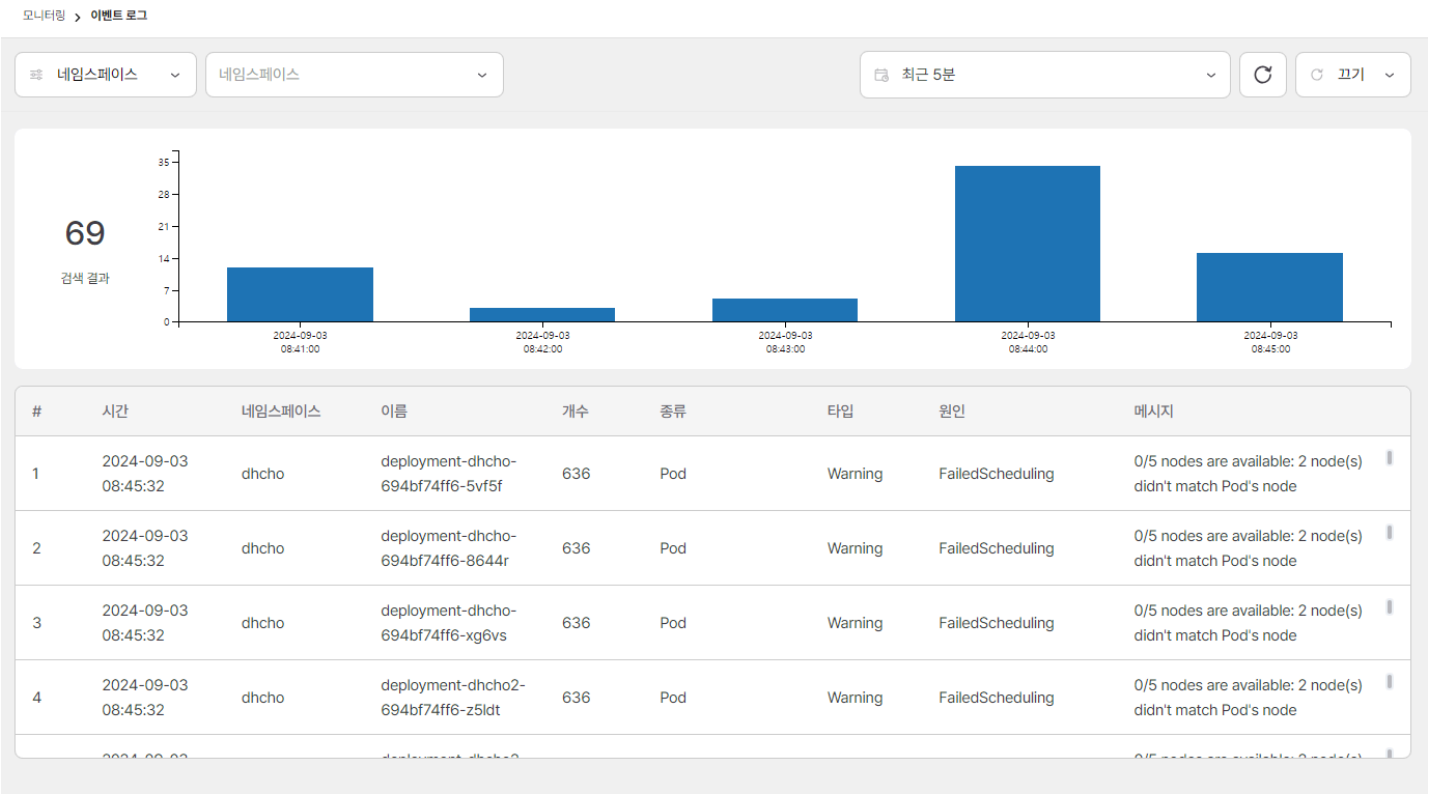

## GPU

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



## 4.2.12.2. 이벤트 로그

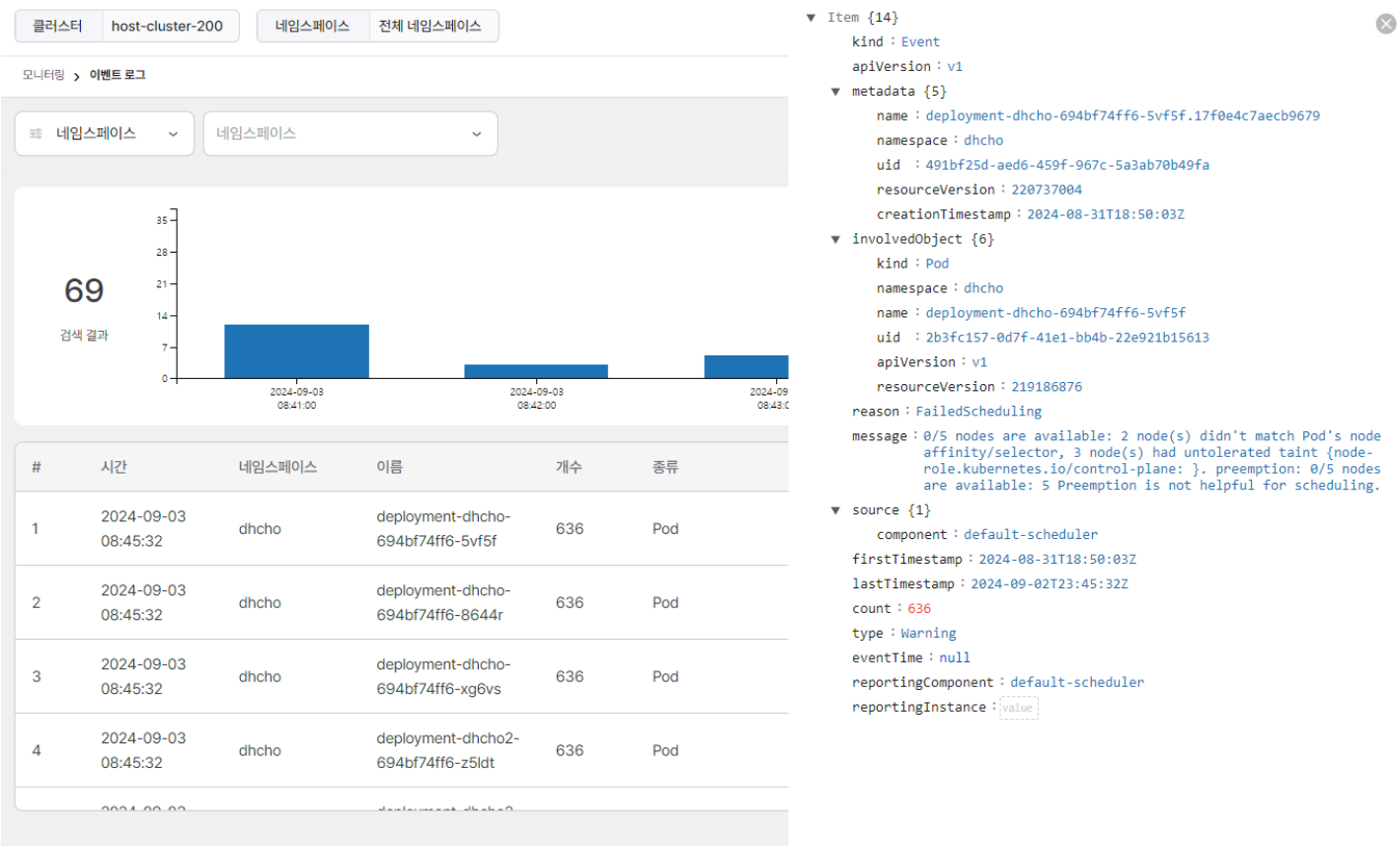
클러스터에서 발생하는 쿠버네티스 이벤트 로그를 제공한다. 시간대별 로그 카운트 정보를 차트로 제공하고 아래 개별 로그에 대한 정보를 제공한다.



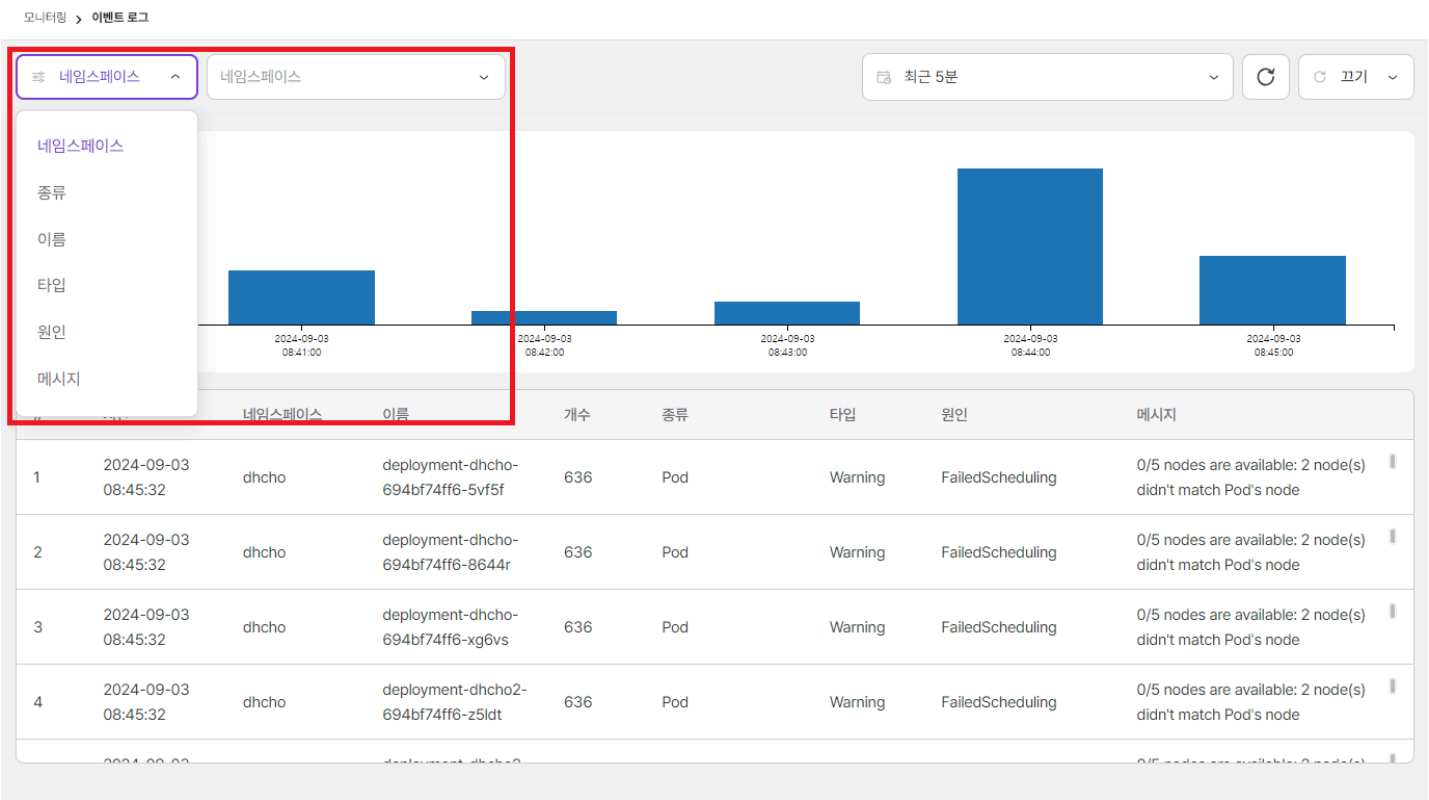

제공하는 정보는 다음과 같다.

| 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.




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.

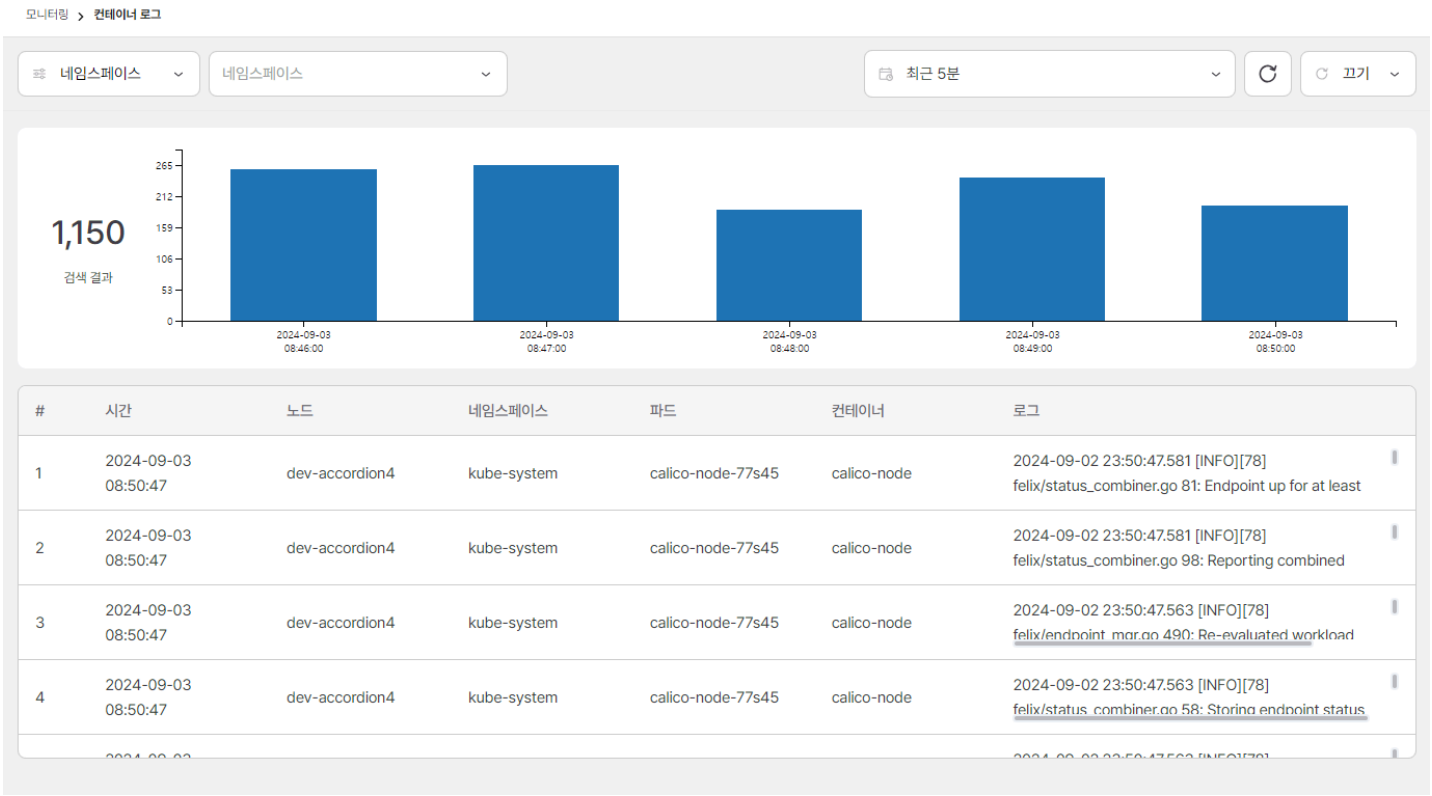


## 4.2.12.3. 컨테이너 로그

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

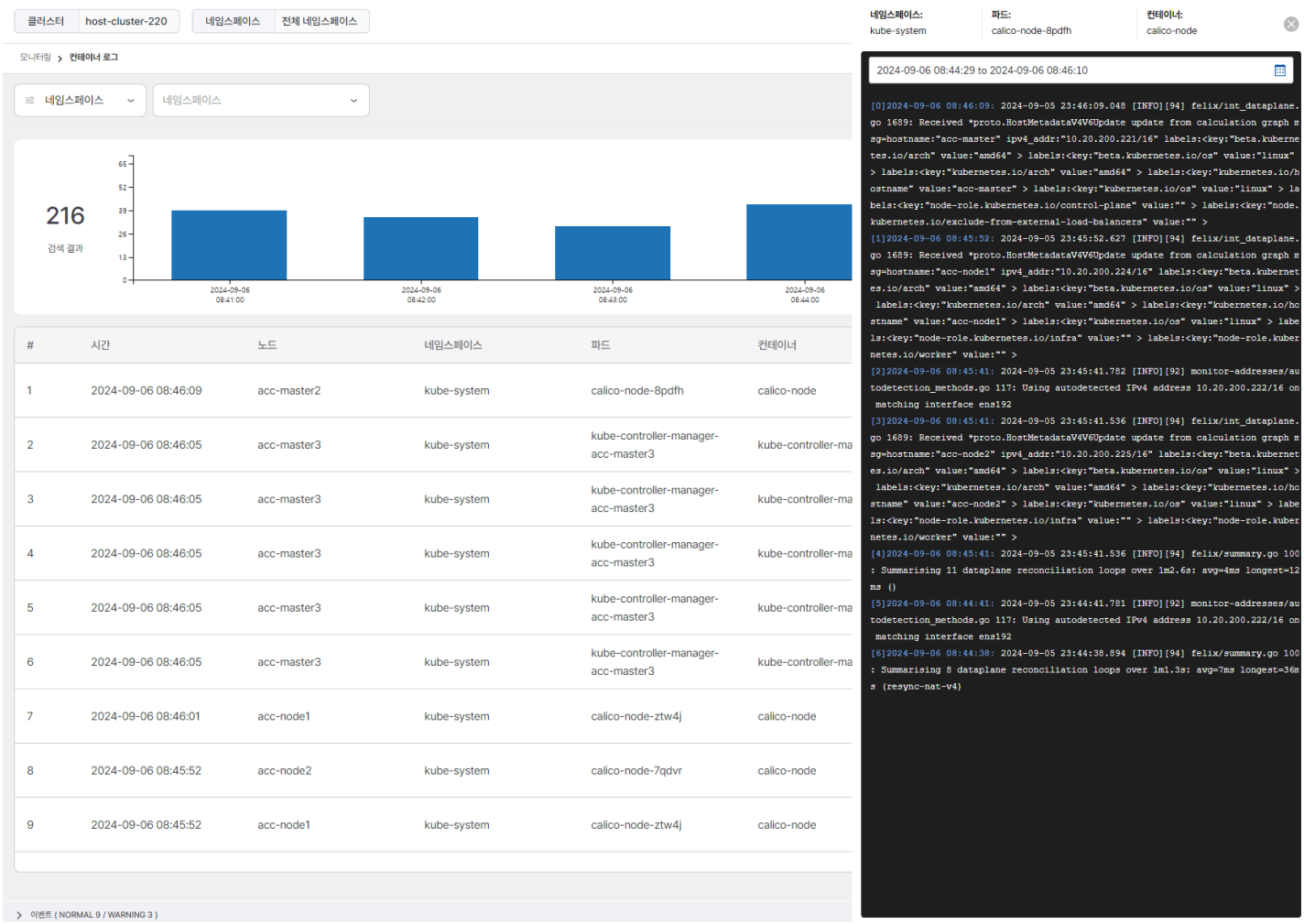
**TIP**

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

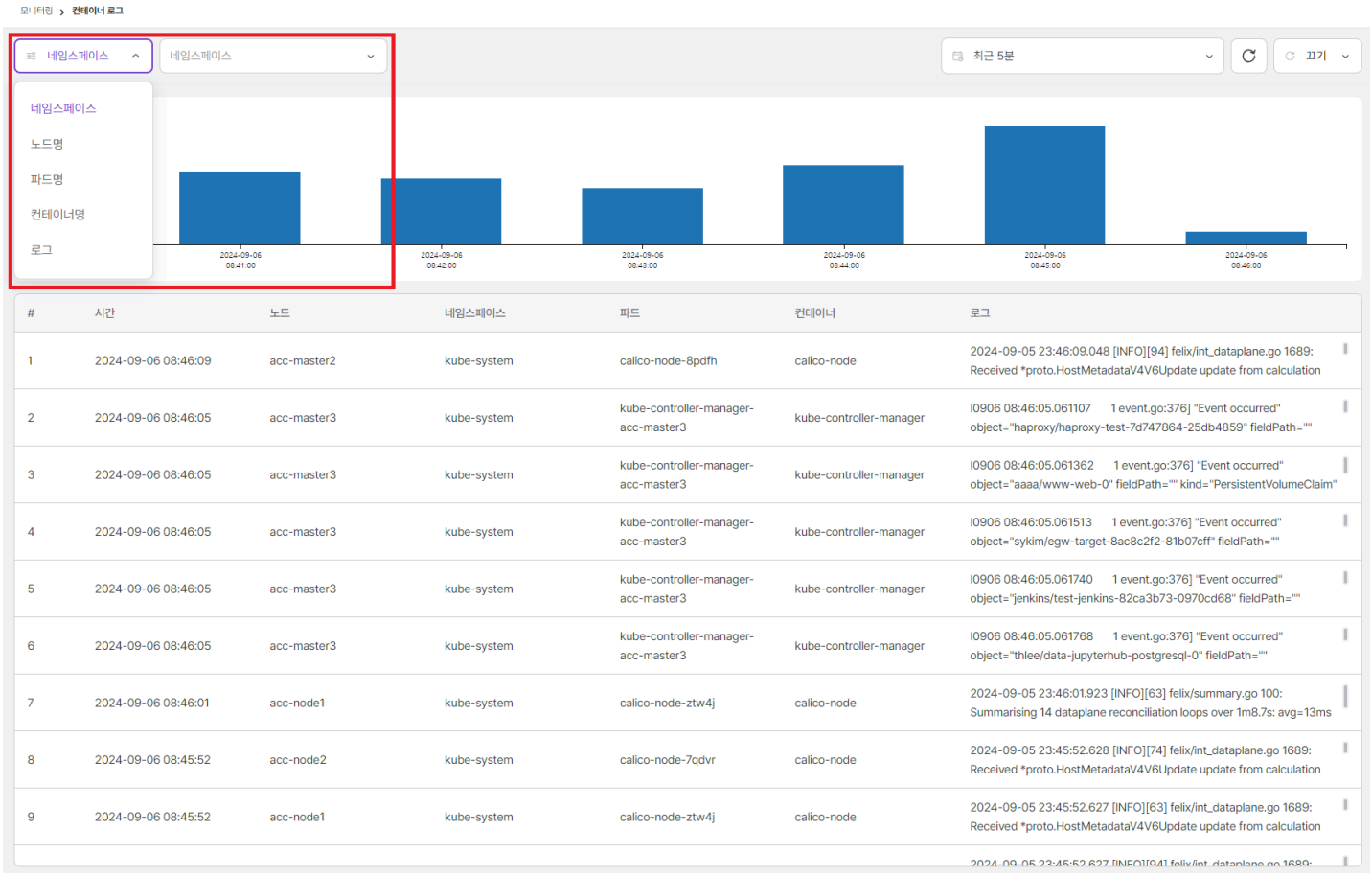



| 항목 | 설명 |
|---|---|
| 시간 | 날짜 (yyyy-mm-dd HH:mm:ss) |
| 노드 | 로그가 발생한 노드 |
| 네임스페이스 | 로그가 발생한 네임스페이스 |
| 파드 | 로그가 발생한 파드 |
| 컨테이너 | 로그가 발생한 컨테이너 |
| 로그 | 로그 메세지 |

로그를 선택하면 상세 정보를 확인할 수 있다.



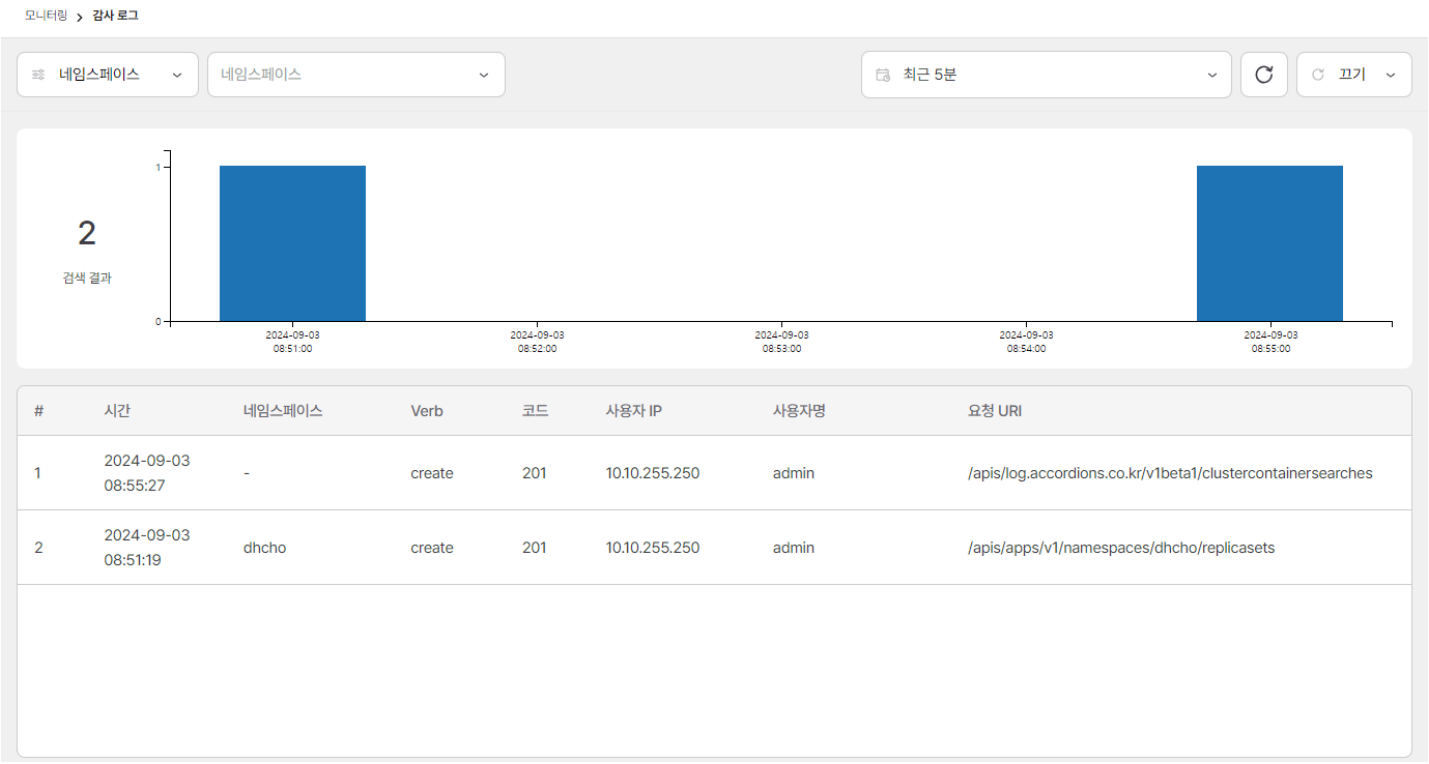

상단의 조건을 설정하여 로그를 필터링할 수 있다. 조건의 항목과 값을 선택하고 돋보기 버튼으로 조건을 설정할 수 있으며 다중 조건 설정이 가능하다. 다중 조건을 설정하면 각각의 조건은 AND 로 조합된다.




| # | 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.HostMetadataV4V6Update update from calculation |
| 2 | 2024-09-06 08:46:05 | acc-master3 | kube-system | kube-controller-manager-acc-master3 | kube-controller-manager | I0906 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 | I0906 08:46:05.061362 1 event.go:376] "Event occurred" object="aaaa/www-web-0" fieldPath="" kind="PersistentVolumeClaim" |
| 4 | 2024-09-06 08:46:05 | acc-master3 | kube-system | kube-controller-manager-acc-master3 | kube-controller-manager | I0906 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 | I0906 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 | I0906 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-7qdvr | calico-node | 2024-09-05 23:45:52.628 [INFO][74] felix/int_dataplane.go 1689: Received *proto.HostMetadataV4V6Update 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.HostMetadataV4V6Update update from calculation |

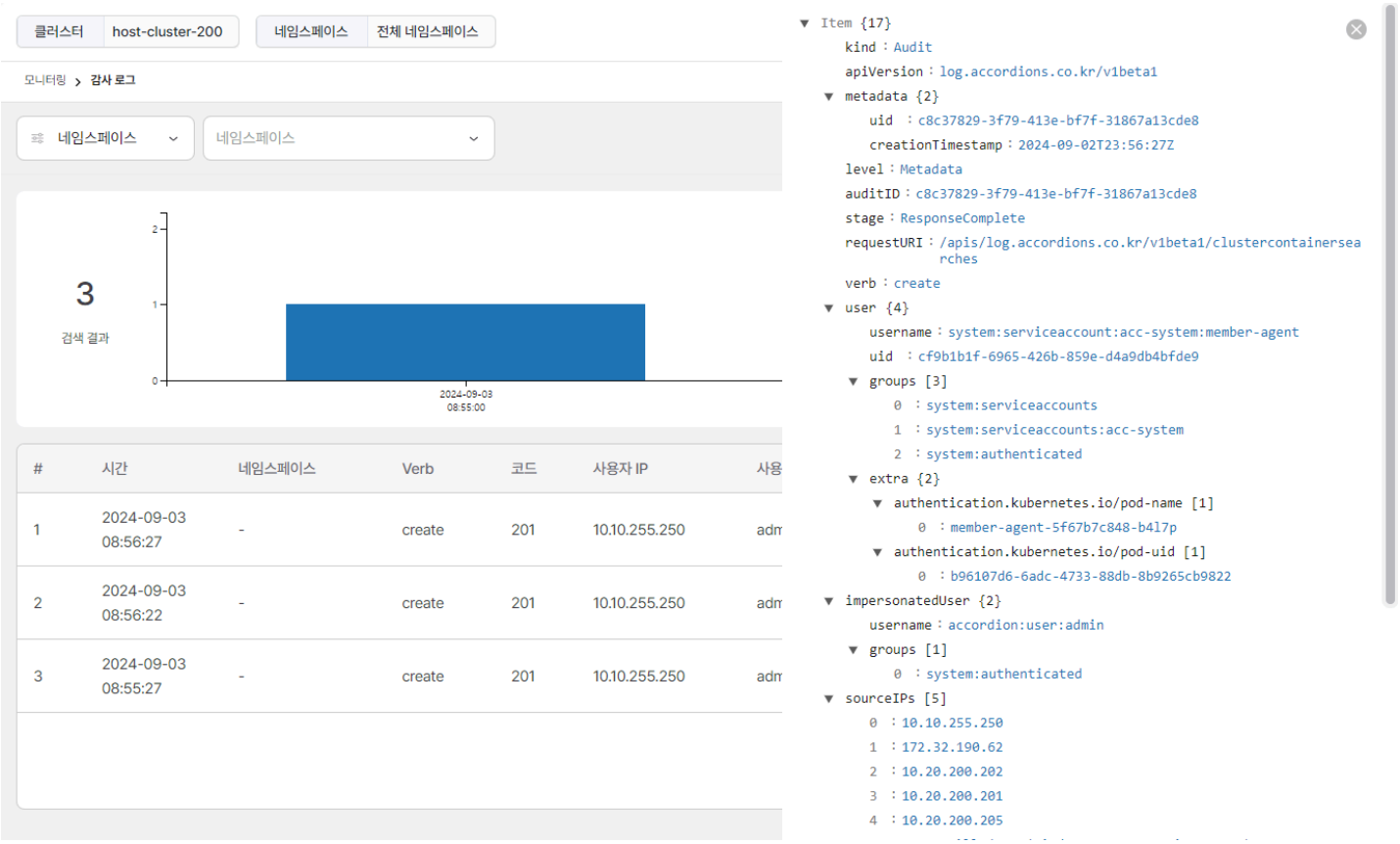
### 4.2.12.4. 감사 로그

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

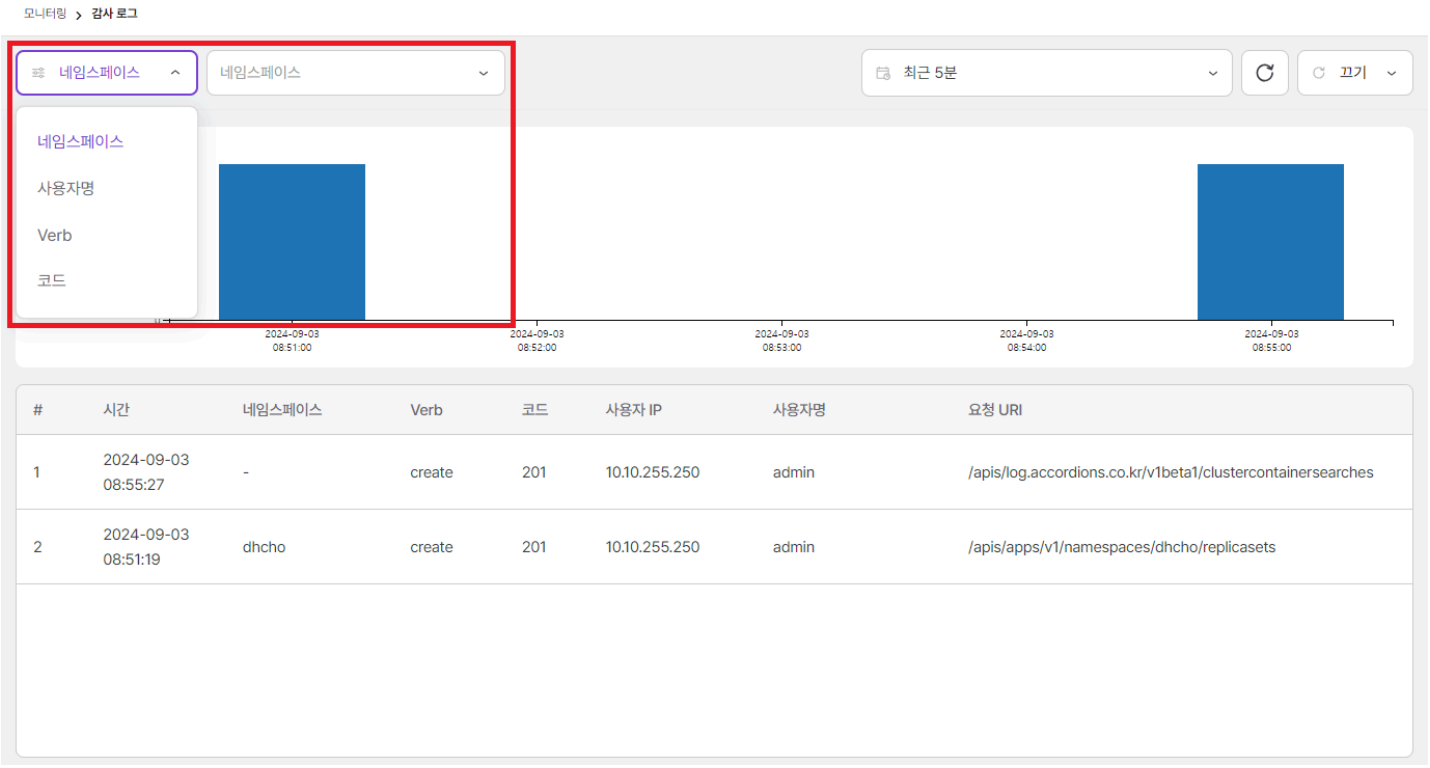



| 항목 | 설명 |
|---|---|
| 시간 | 날짜 (yyyy-mm-dd HH:mm:ss) |
| 네임스페이스 | 사용자가 접근한 리소스가 배포된 네임스페이스 |
| VERB | 리소스 접근 요청시 사용한 API 동사 (get, list, create, update, patch, watch, delete) |
| 코드 | 리소스 접근 요청에 대한 응답 HTTP 코드 |
| 사용자 IP | 요청이 발생한 IP |
| 사용자명 | 리소스 접근 요청을 한 사용자 |
| 요청 URI | 리소스 접근 요청 엔드포인트 |

로그를 선택하면 상세 정보를 확인할 수 있다.

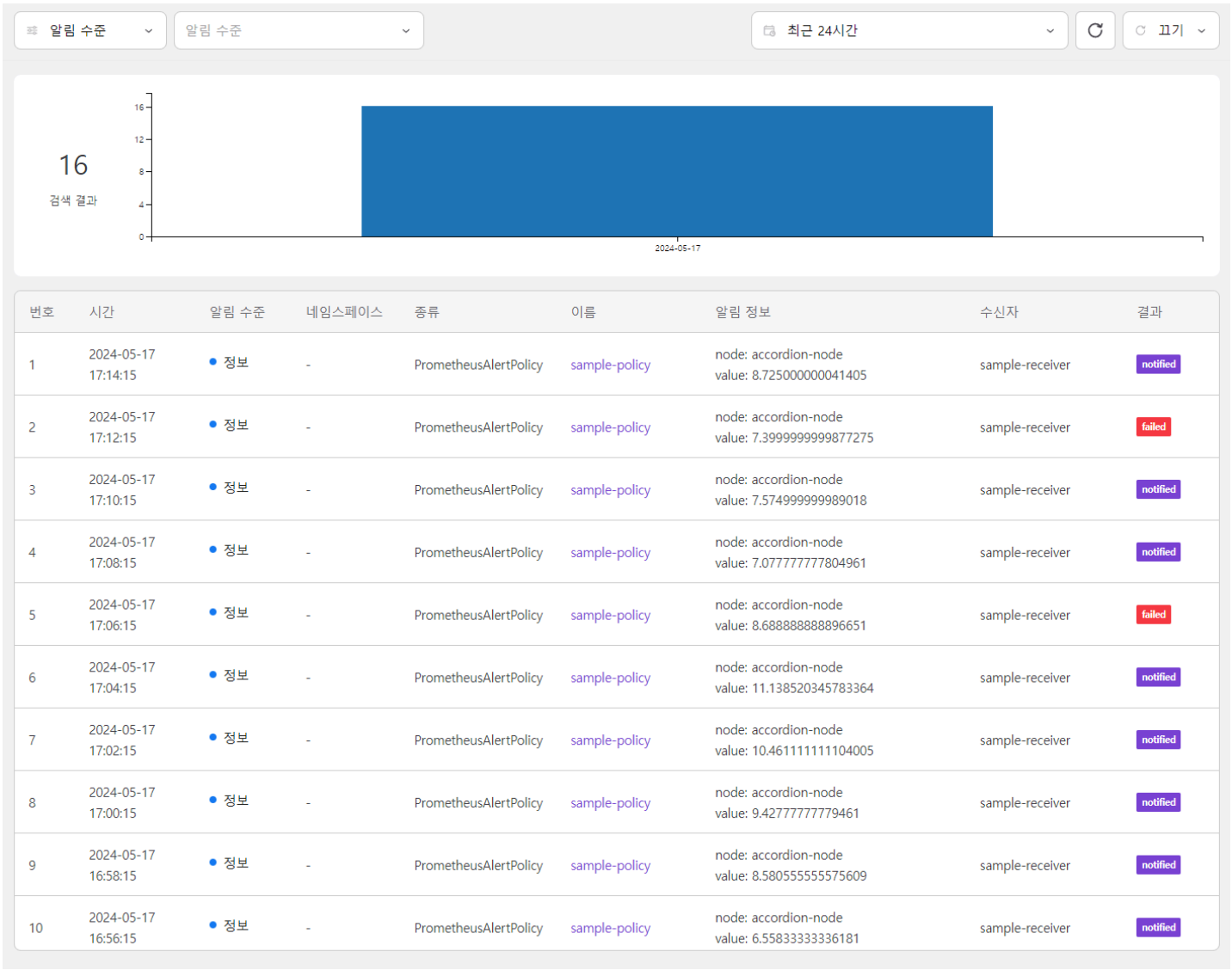



상단의 조건을 설정하여 로그를 필터링할 수 있다. 조건의 항목과 값을 선택하고 돋보기 버튼으로 조건을 설정할 수 있으며 다중 조건 설정이 가능하다. 다중 조건을 설정하면 각각의 조건은 AND 로 조합된다.



### 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.



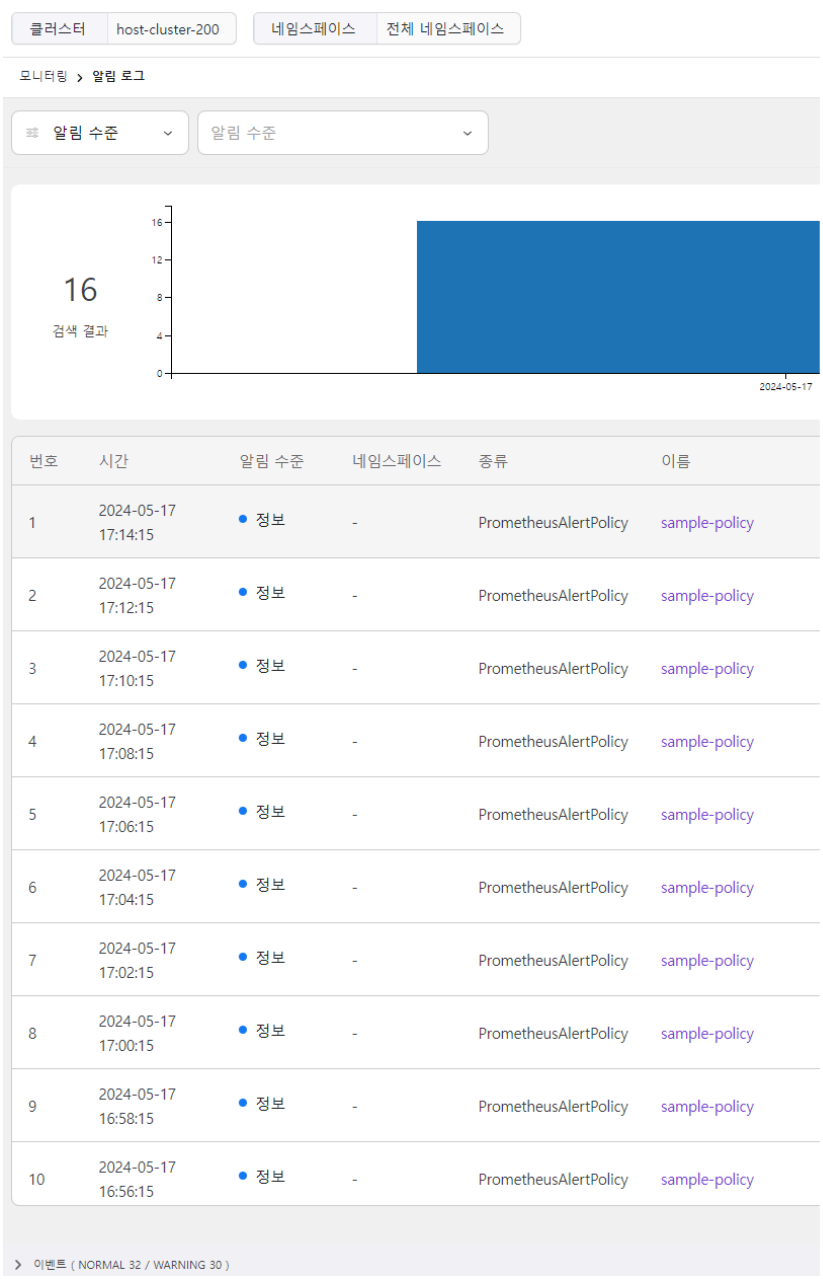

| 번호 | 시간 | 알림 수준 | 네임스페이스 | 종류 | 이름 | 알림 정보 | 수신자 | 결과 |
|---|---|---|---|---|---|---|---|---|
| 1 | 2024-05-17 17:14:15 | 정보 | - | PrometheusAlertPolicy | sample-policy | node: accordion-node value: 8.725000000041405 | sample-receiver | notified |
| 2 | 2024-05-17 17:12:15 | 정보 | - | PrometheusAlertPolicy | sample-policy | node: accordion-node value: 7.3999999999877275 | sample-receiver | failed |
| 3 | 2024-05-17 17:10:15 | 정보 | - | PrometheusAlertPolicy | sample-policy | node: accordion-node value: 7.574999999989018 | sample-receiver | notified |
| 4 | 2024-05-17 17:08:15 | 정보 | - | PrometheusAlertPolicy | sample-policy | node: accordion-node value: 7.077777777804961 | sample-receiver | notified |
| 5 | 2024-05-17 17:06:15 | 정보 | - | PrometheusAlertPolicy | sample-policy | node: accordion-node value: 8.688888888896651 | sample-receiver | failed |
| 6 | 2024-05-17 17:04:15 | 정보 | - | PrometheusAlertPolicy | sample-policy | node: accordion-node value: 11.138520345783364 | sample-receiver | notified |
| 7 | 2024-05-17 17:02:15 | 정보 | - | PrometheusAlertPolicy | sample-policy | node: accordion-node value: 10.461111111104005 | sample-receiver | notified |
| 8 | 2024-05-17 17:00:15 | 정보 | - | PrometheusAlertPolicy | sample-policy | node: accordion-node value: 9.42777777779461 | sample-receiver | notified |
| 9 | 2024-05-17 16:58:15 | 정보 | - | PrometheusAlertPolicy | sample-policy | node: accordion-node value: 8.580555555575609 | sample-receiver | notified |
| 10 | 2024-05-17 16:56:15 | 정보 | - | PrometheusAlertPolicy | sample-policy | node: accordion-node value: 6.55833333336181 | sample-receiver | notified |

제공하는 정보는 다음과 같다.

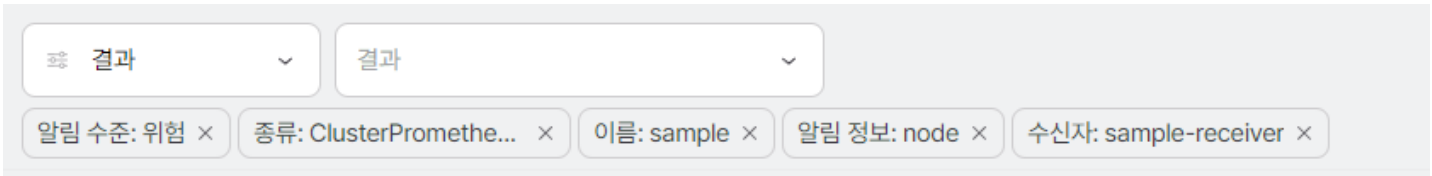
| 항목 | 설명 |
|---|---|
| 시간 | 날짜 (yyyy-mm-dd HH:mm:ss) |
| 알림 수준 | 알림 경보 수준 |
| 네임스페이스 | 네임스페이스 이름 |
| 종류 | 알림 발생 리소스 종류 |
| 이름 | 알림 발생 리소스 이름 |
| 알림 정보 | 알림 발생 정보 |
| 수신자 | 알림을 받는 수신자 목록 |

| 항목 | 설명 |
| --- | --- |
| 결과 | 알림 발송 결과 |

로그를 선택하면 상세 정보를 확인할 수 있다.

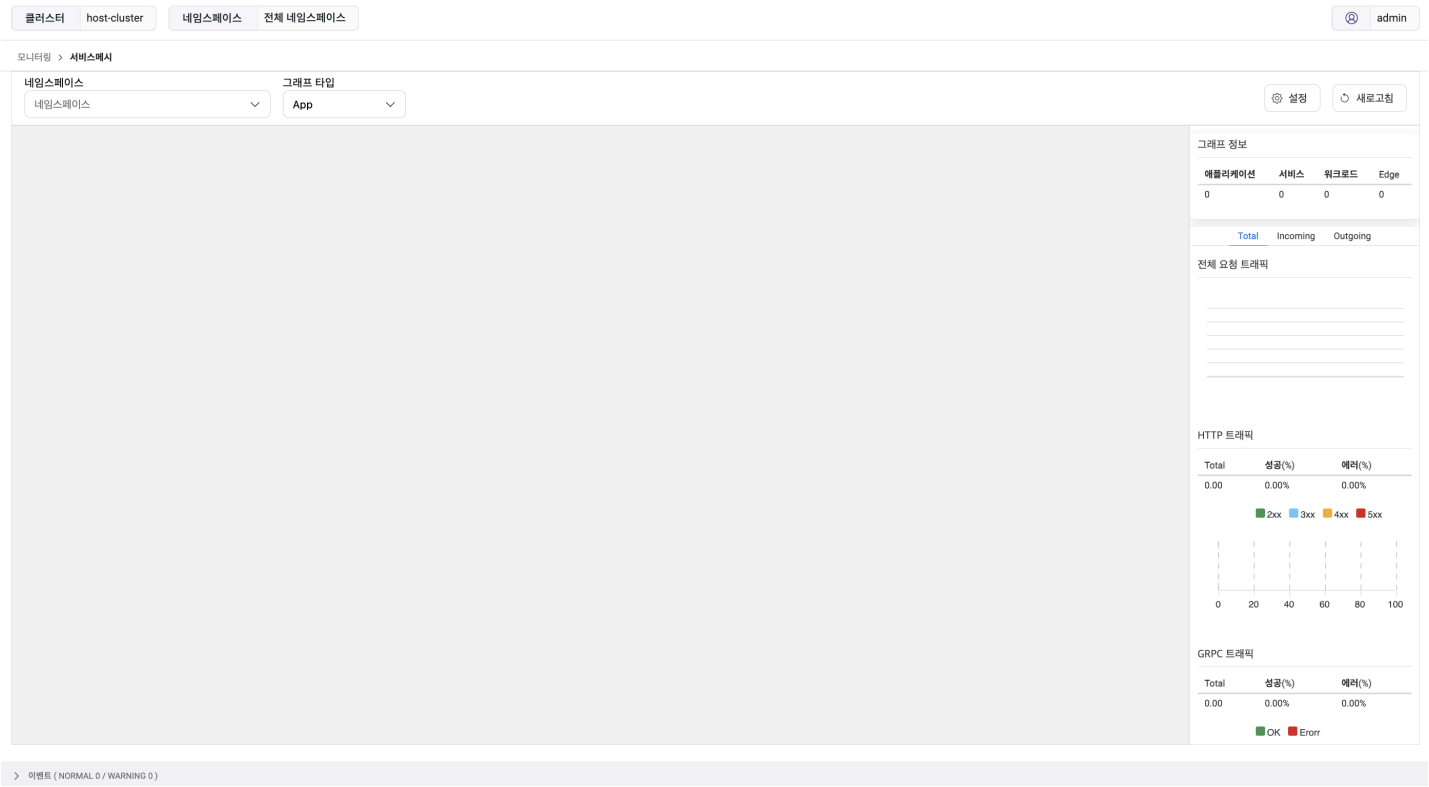


상단의 좌측의 조건을 설정하여 로그를 필터링할 수 있다. 조건의 항목과 값을 선택 및 `검색` 버튼으로 추가할 수 있으며 다중 조건 설정이 가능하다. 다중 조건을 설정하면 각각의 조건은 `AND` 로 조합된다.



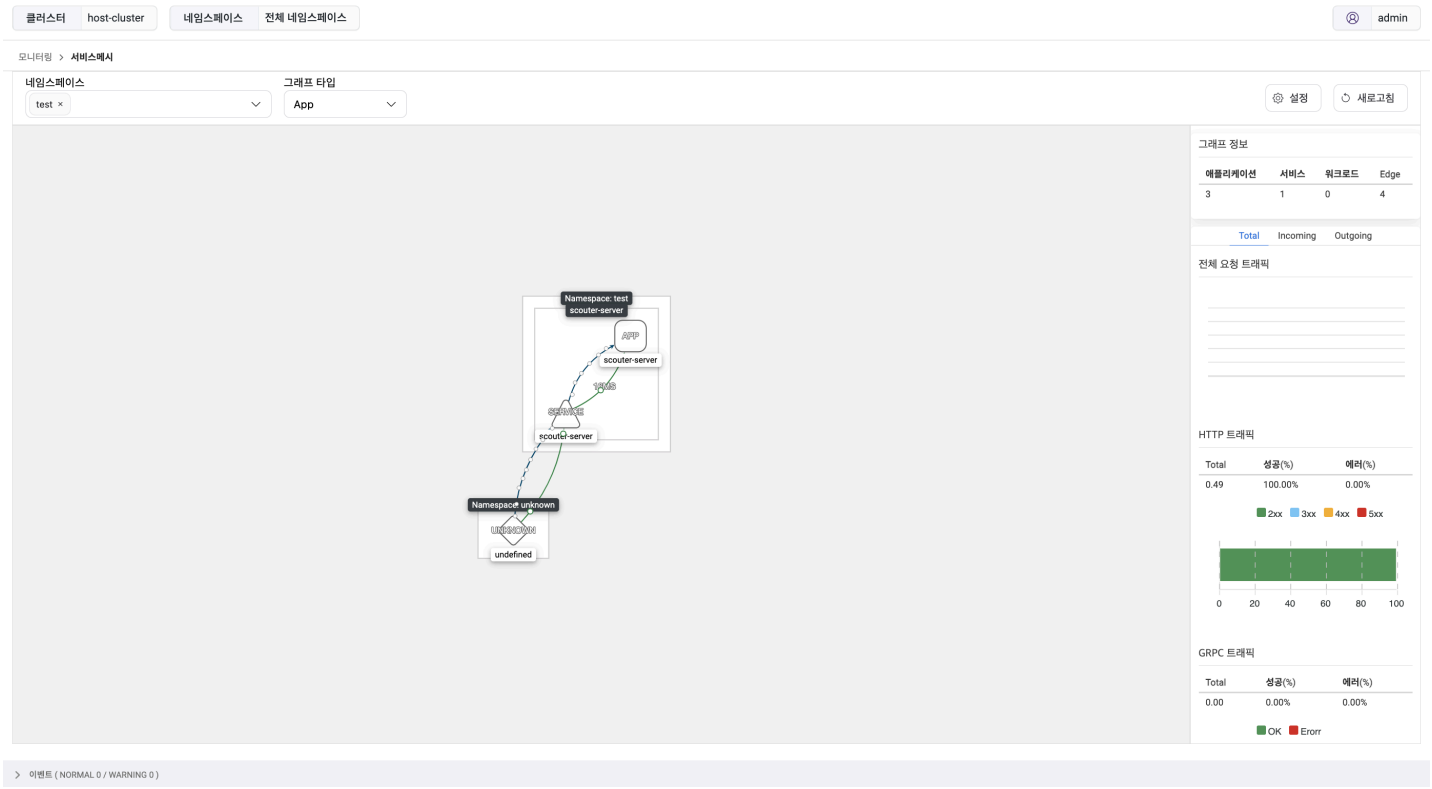
## 4.2.12.6. 서비스메시

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

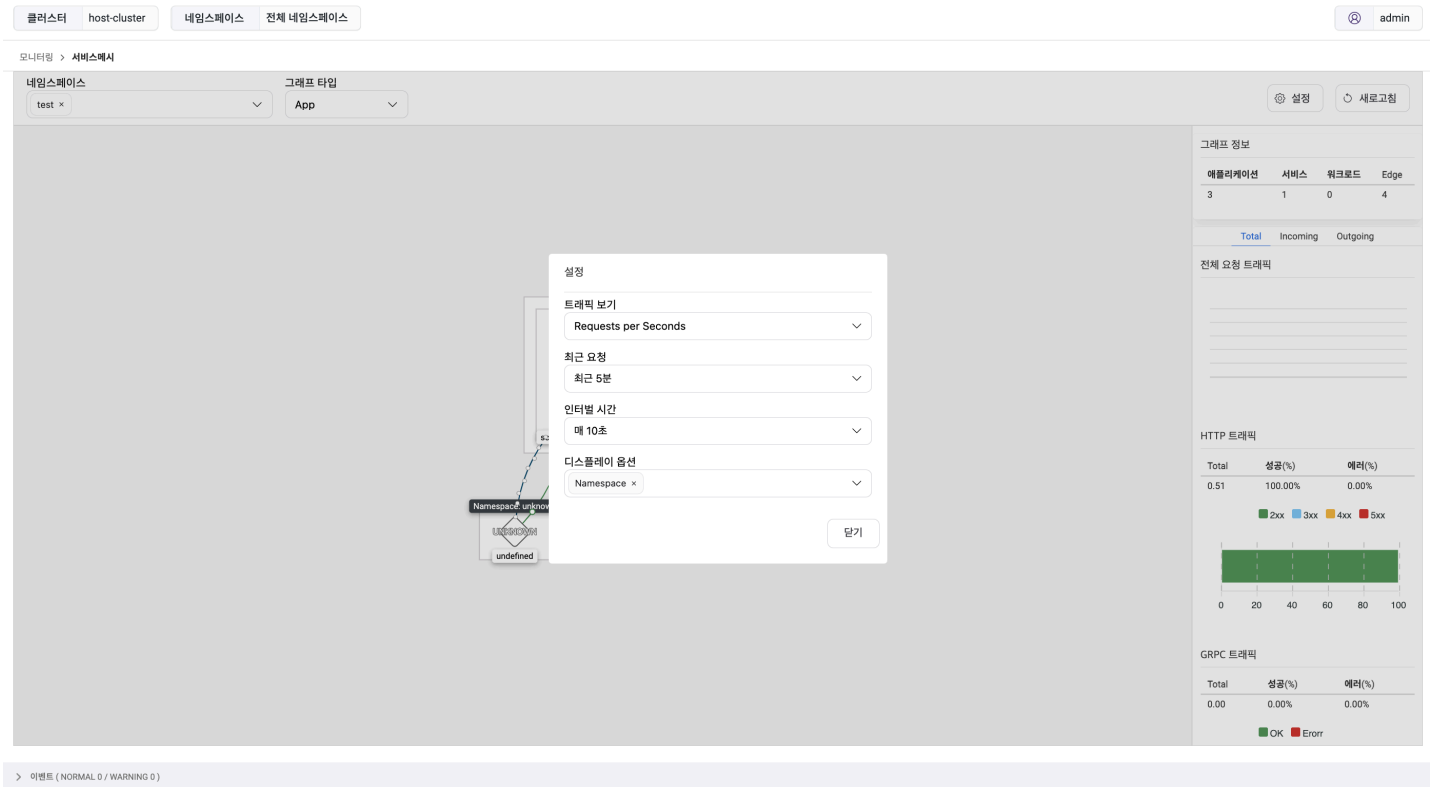



상단의 네임스페이스 항목에 모니터링을 원하는 네임스페이스 목록을 선택하면 해당 네임스페이스 내에서 발생하는 서비스 간의 통신 상태를 그래프로 확인할 수 있고 그래프 타입을 변경하여 관점의 기준을 애플리케이션이나 애플리케이션 버전, 서비스, 워크로드 등으로 변경할 수 있다.

**TIP** 네임스페이스는 다중 선택이 가능하다.




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

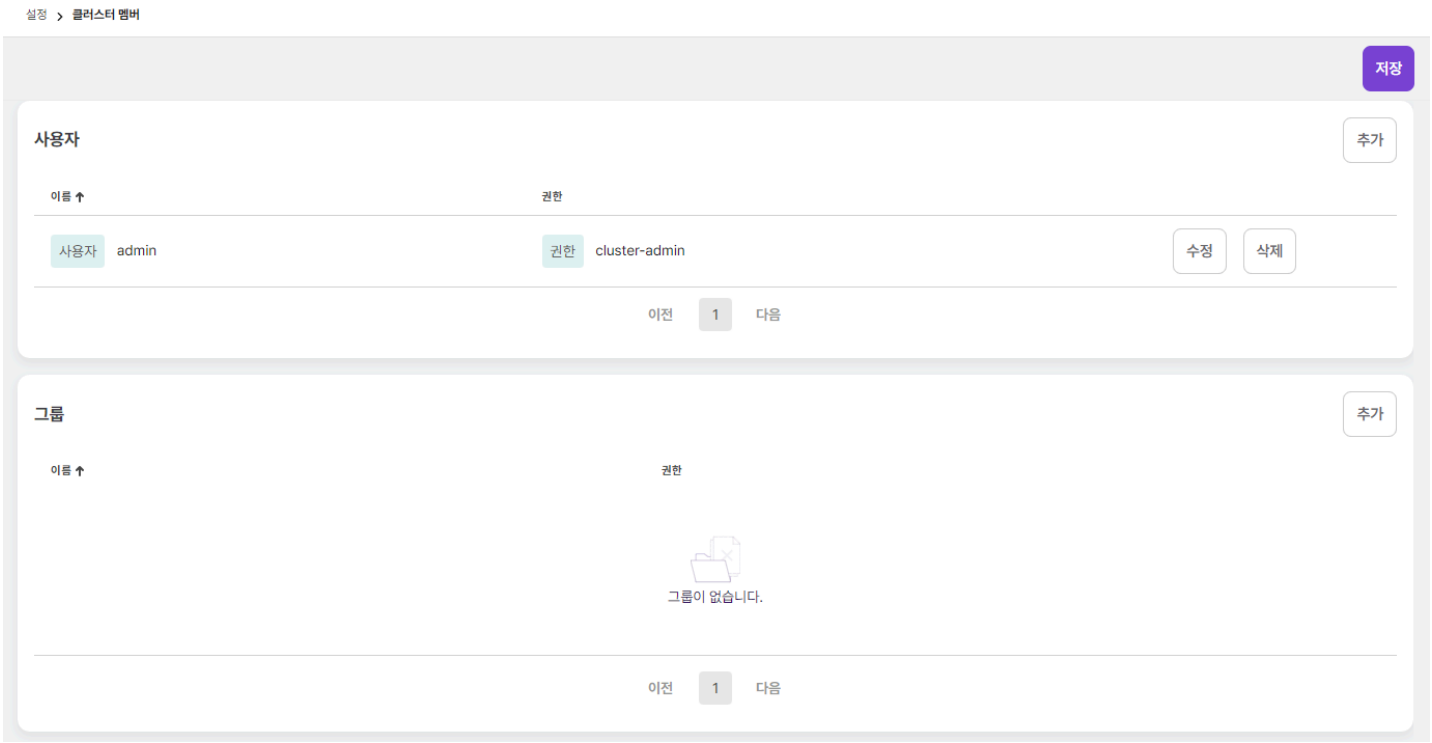


## 4.2.13. 설정

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

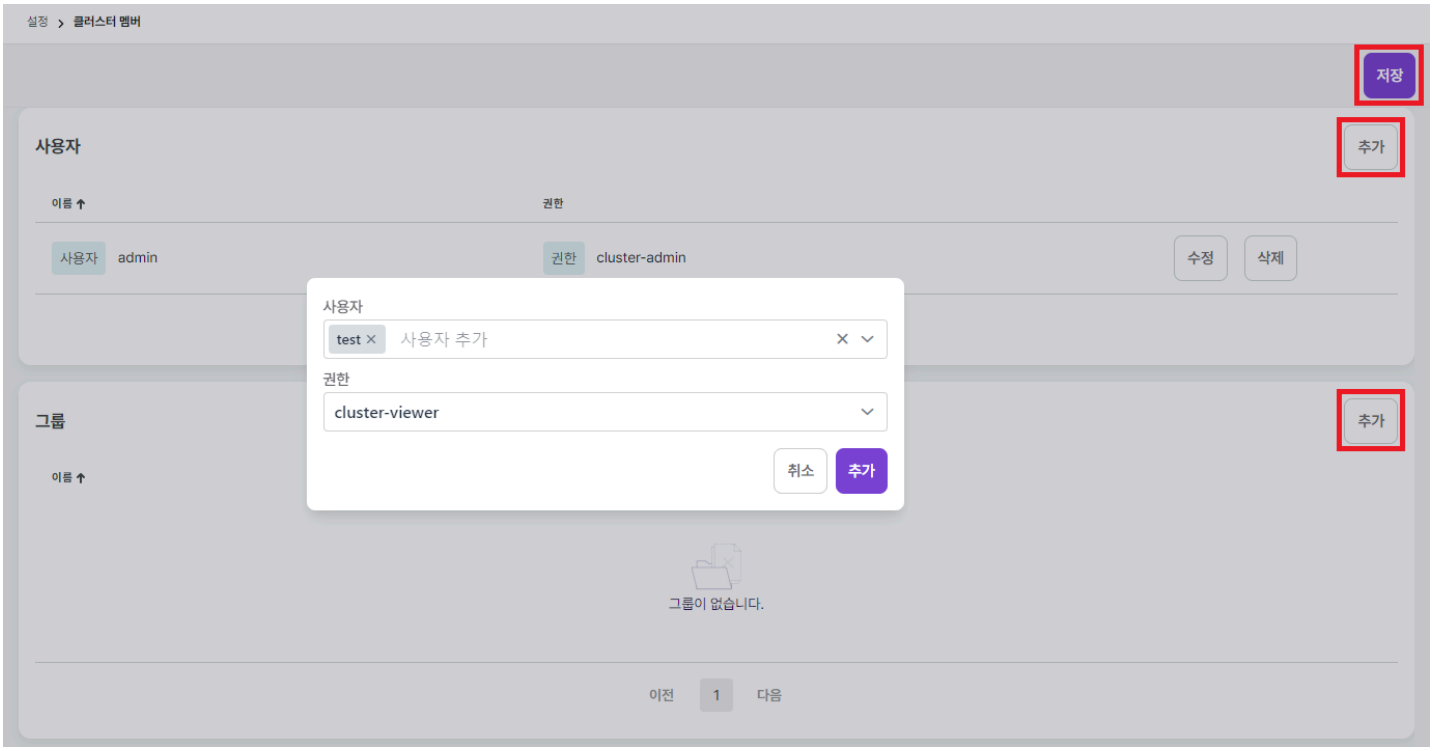
### 4.2.13.1. 클러스터 멤버

클러스터 사용자와 그룹의 권한을 관리한다.



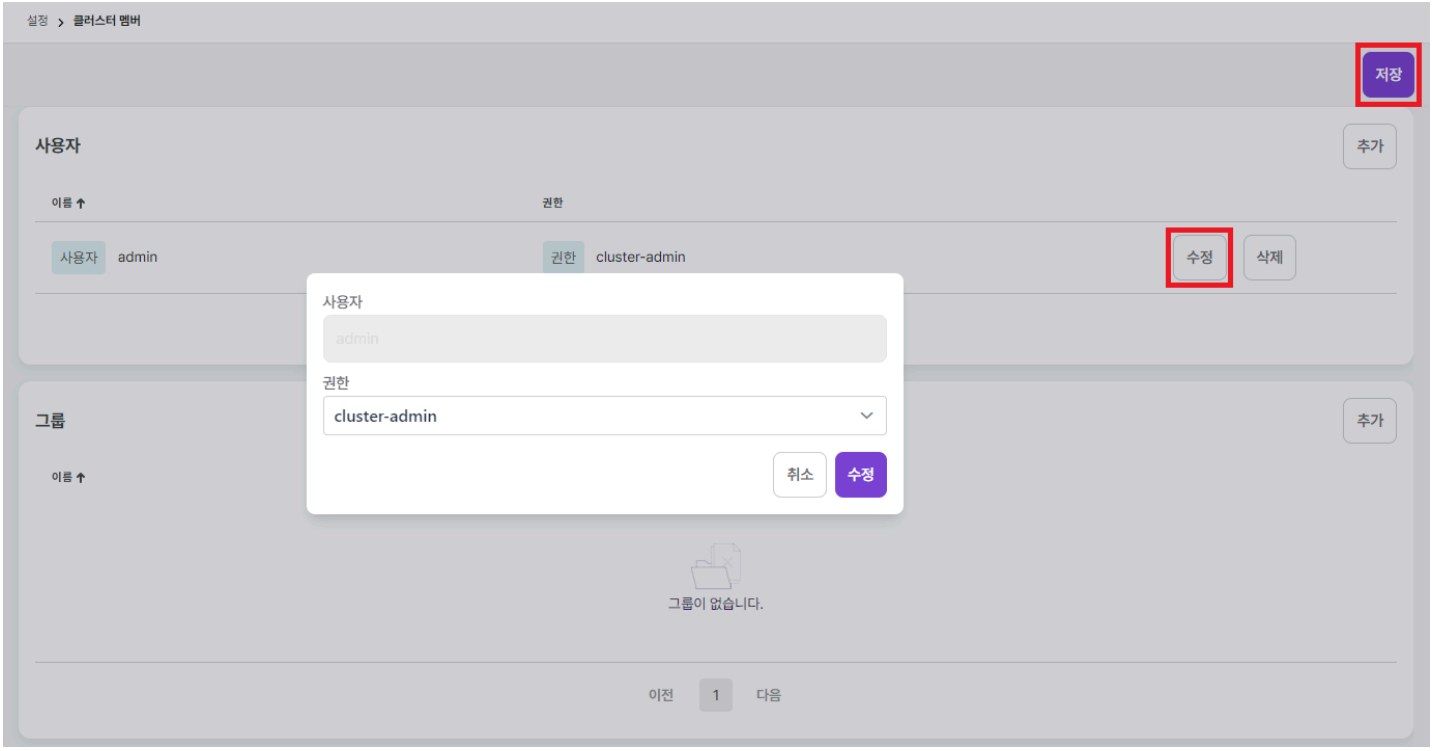

#### 클러스터 멤버 추가

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.



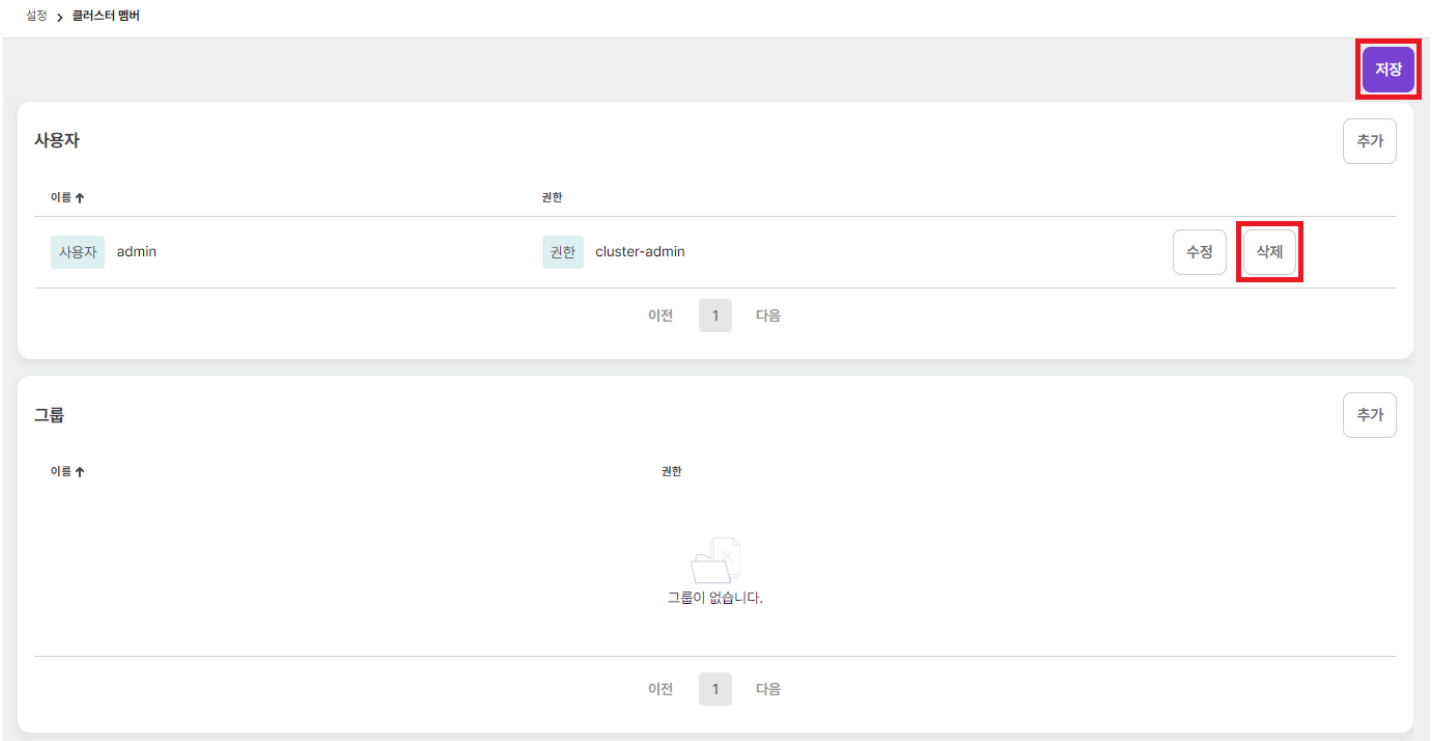
## 클러스터 멤버 수정

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.



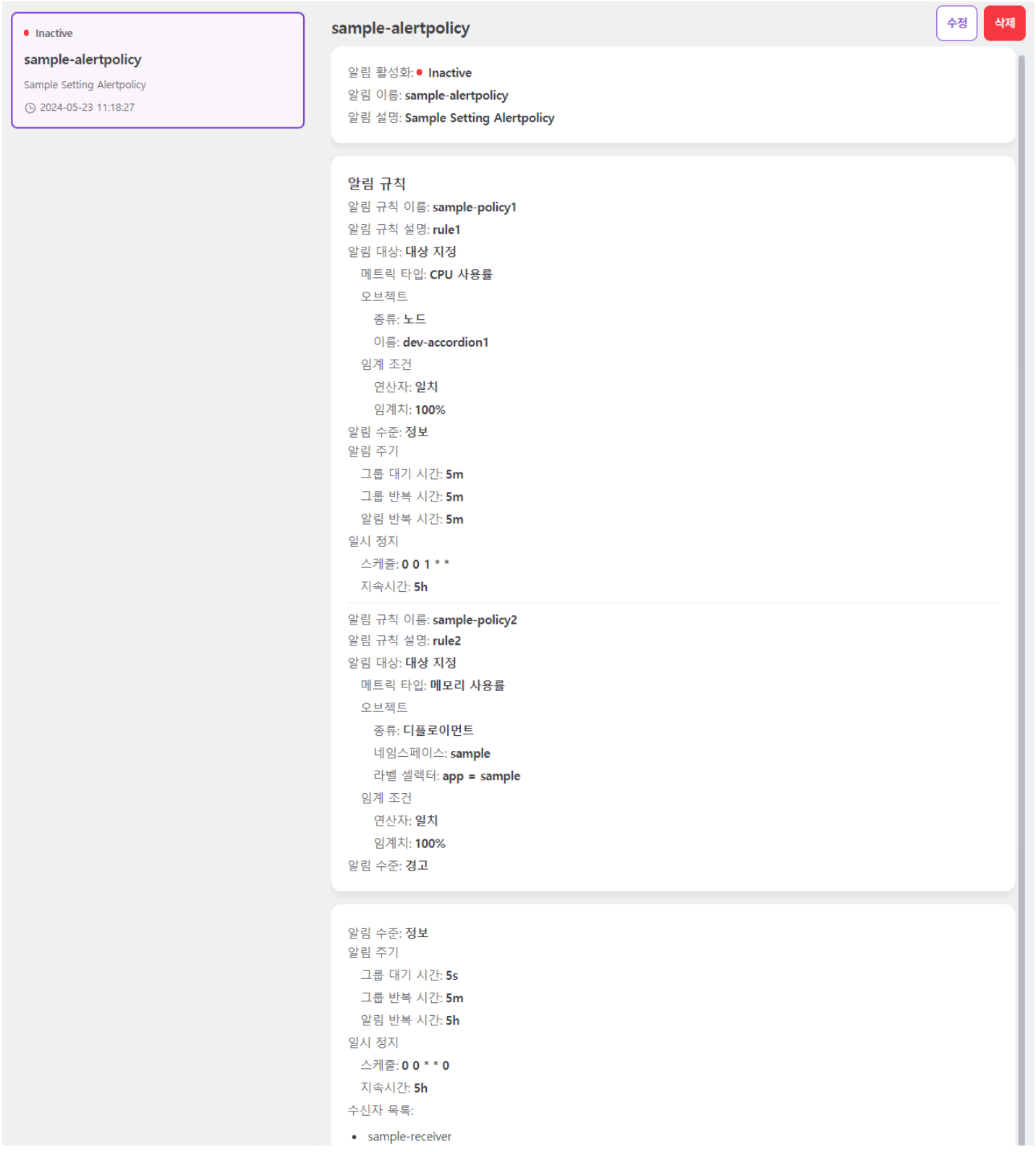
### 4.2.13.2. 클러스터 수신자

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 되는경우 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.

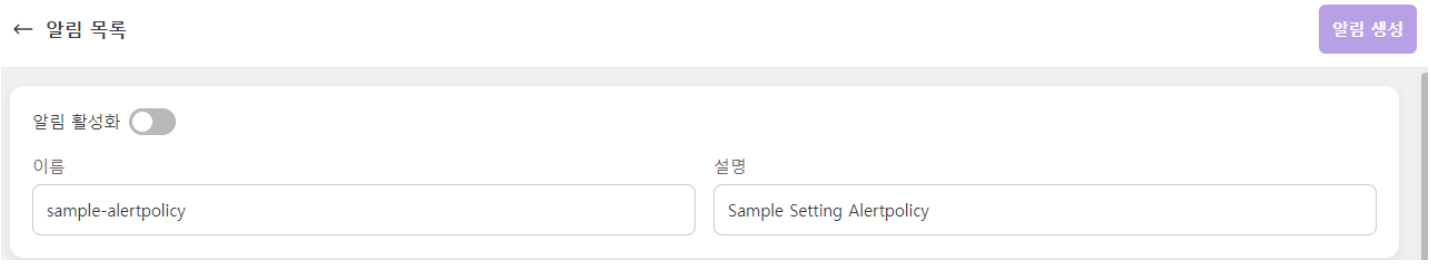
**TIP** 알림 정책의 기준은 사용자가 임의로 정의하여 설정할 수 있다.



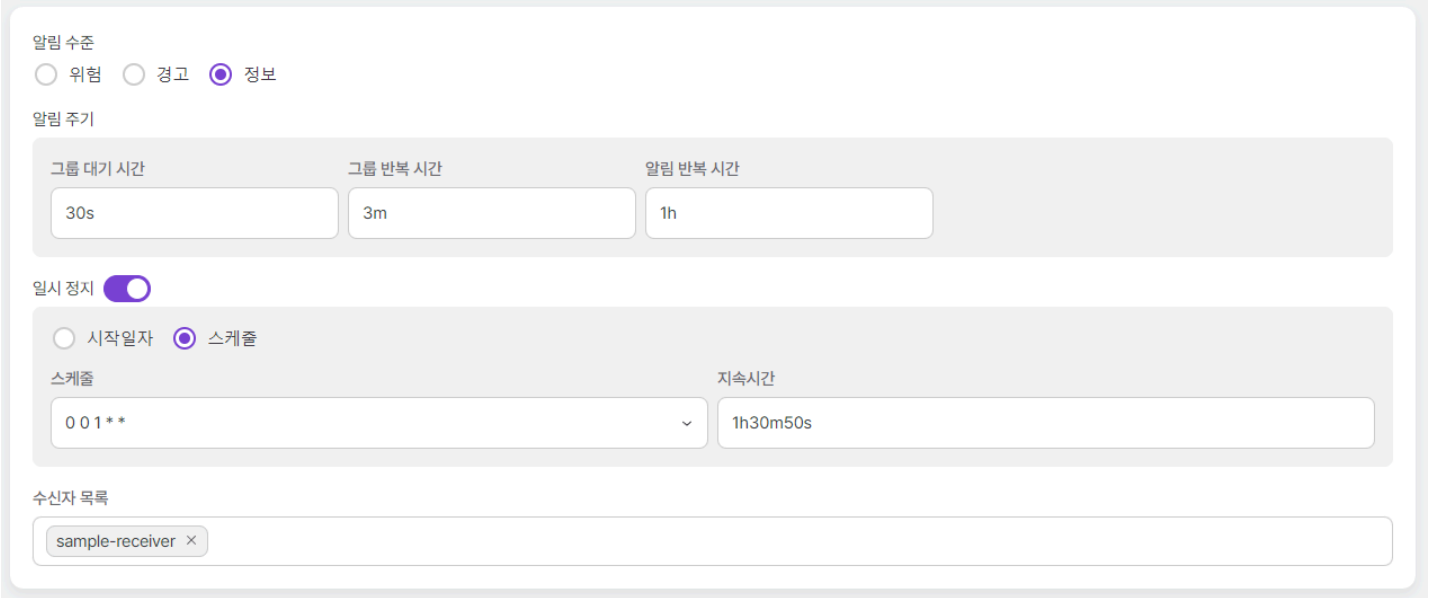
## 알림 정책 생성

+ 알림 생성 버튼을 선택하면 알림을 설정할 수 있는 폼이 나타난다. 폼에서 설정 정보를 입력하고 알림 생성 를 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.




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



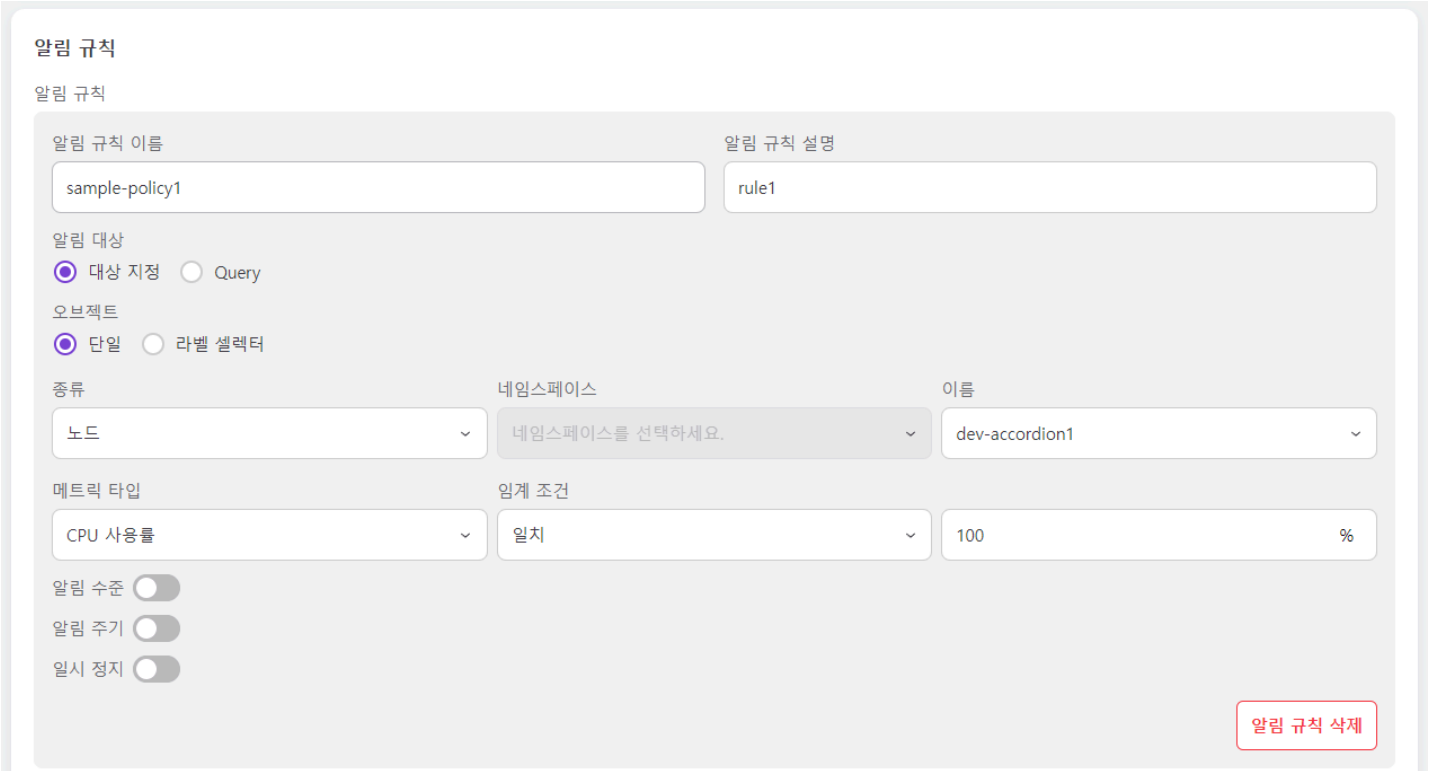

| 항목 | 설명 |
| --- | --- |
| 알림 수준 | 알림 경보 수준 |
| 알림 주기 | 알림 발송을 위한 시간을 설정<br>• 그룹 대기 시간: 처음 알림이 대기하는 시간<br>• 그룹 반복 시간: 알림 발송 후 재발송되기까지 대기 시간<br>• 알람 반복 시간: 알림 재발송 시간 간격 |

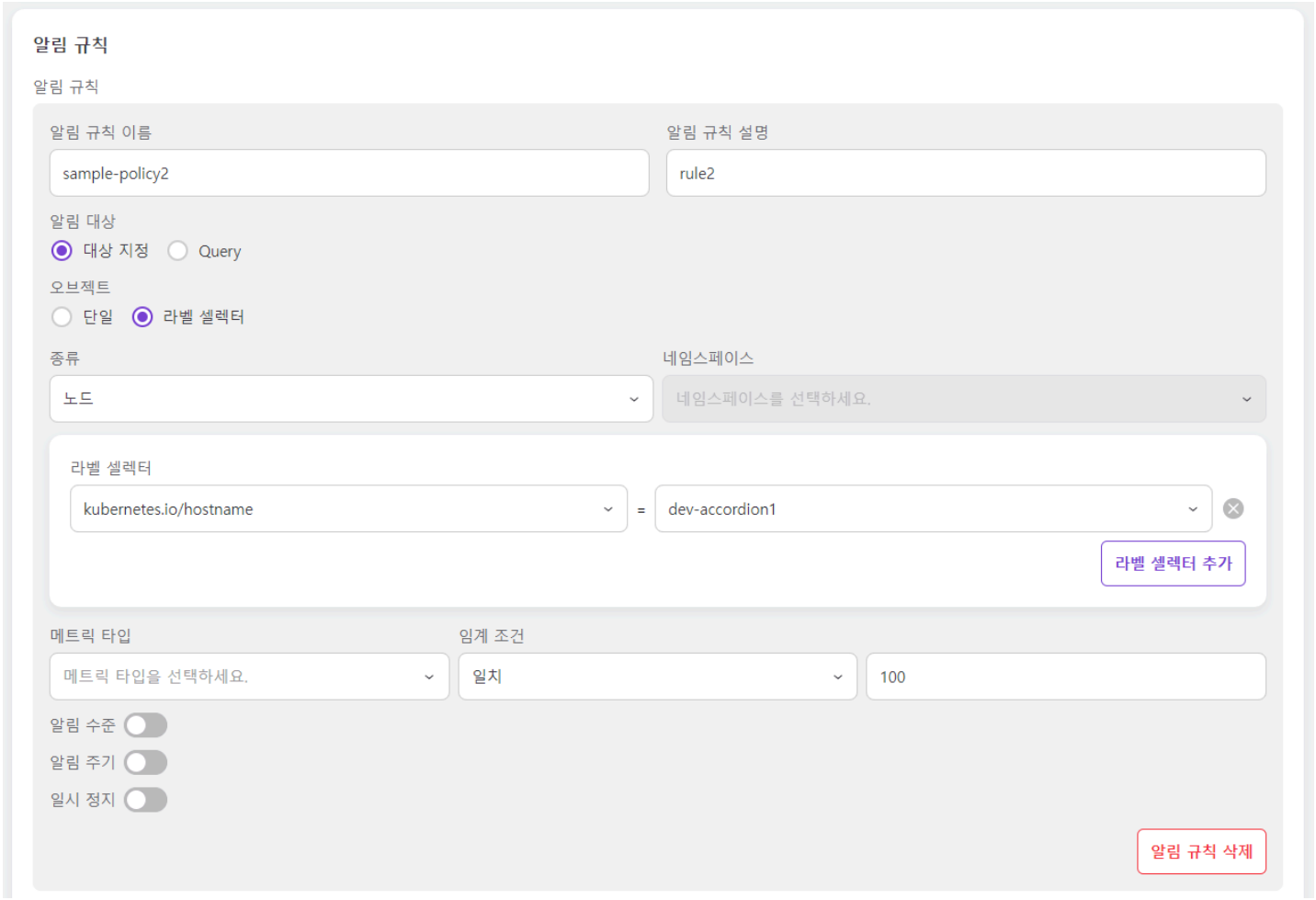
| 항목 | 설명 |
| --- | --- |
| 일시 정지 | 시작일자 혹은 스케줄 을 기준으로 지속 시간 만큼 알림 발송을 제한 |
| 수신자 목록 | 알림을 받을 클러스터 수신자 목록 (클러스터 수신자 메뉴에서 생성) |

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<br>• 그룹 대기 시간: 처음 알림이 대기하는 시간<br>• 그룹 반복 시간: 알림 발송 후 재발송되기까지 대기 시간<br>• 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



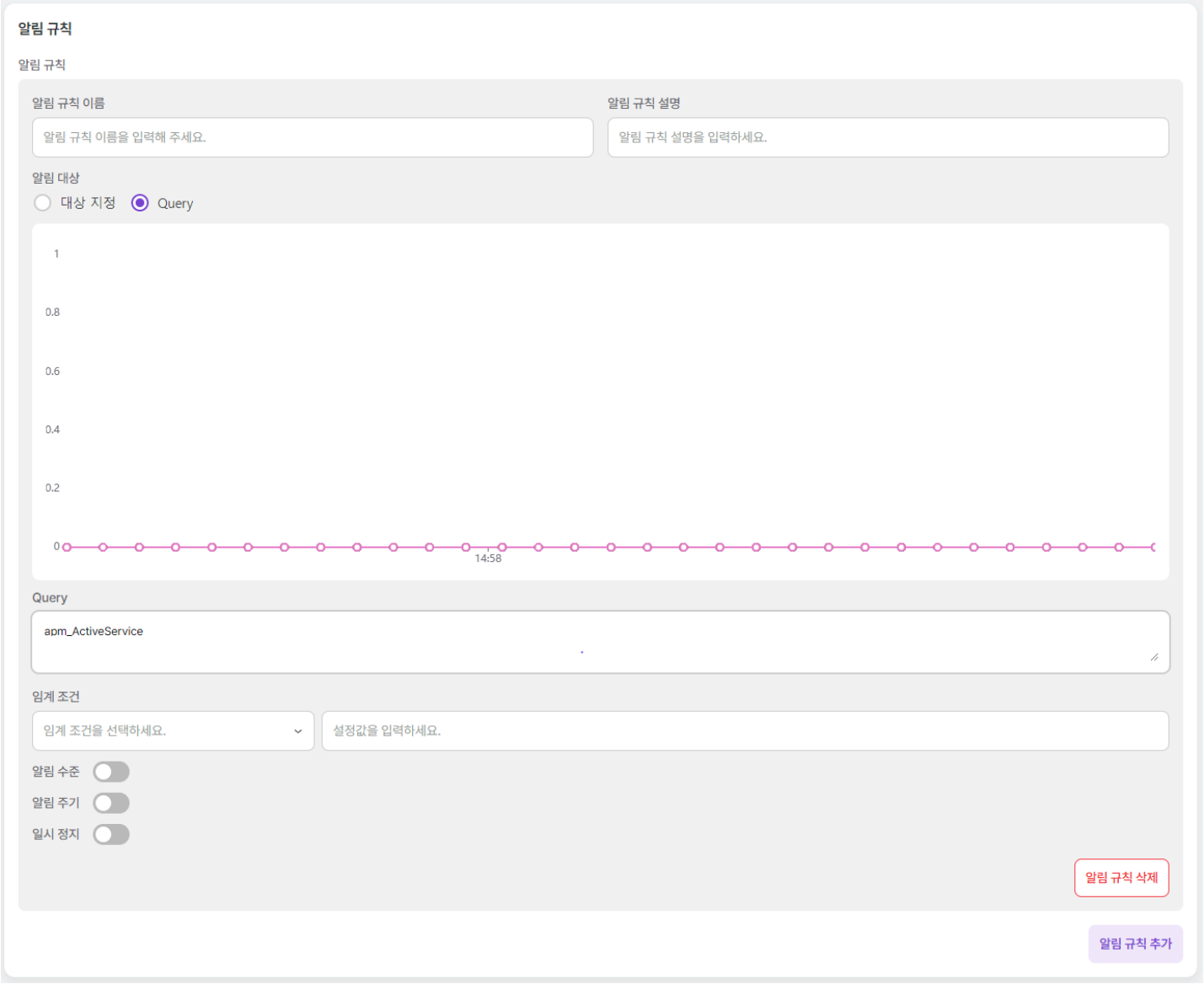



| **오브젝트** | 단일 **혹은** 라벨 셀렉터 **선택** |
|---|---|
| 종류 | 노드 혹은 워크로드(디플로이먼트, 스테이트풀셋, 데몬셋, 잡, 크론잡, 파드) 중 선택 |
| 네임스페이스 | 네임스페이스 선택 |
| 이름 | 오브젝트가 단일 의 경우 리소스 이름 선택 |
| 라벨 셀렉터 | 오브젝트가 라벨 셀렉터 의 경우 라벨 셀렉터 설정<br>• 키: 매칭 할 라벨 키 값<br>• 밸류: 매칭 할 라벨 벨류 값 |

| **오브젝트** | 단일 **혹은** 라벨 셀렉터 **선택** |
| --- | --- |
| 메트릭 타입 | • 노드 메트릭 타입<br>  • CPU 사용률<br>  • 메모리 사용률<br>  • 디스크 사용률<br>• 워크로드 메트릭 타입<br>  • CPU 사용률<br>  • 메모리 사용률<br>  • 액티브 서비스<br>  • 응답시간<br>  • Gc 시간 |
| 임계 조건 | • 일치<br>• 불일치<br>• 보다 크다<br>• 크거나 같다<br>• 보다 작다<br>• 작거나 같다 |

| **오브젝트** | 단일 **혹은** 라벨 셀렉터 **선택** |
| --- | --- |
| | 임계 조건의 값 입력<br>• % (Percent)<br>  • CPU 사용률<br>  • 메모리 사용률<br>  • 디스크 사용률<br>• 개수 (Count)<br>  • 액티브 서비스<br>• 초 (Second)<br>  • 응답시간<br>  • Gc 시간 |

- 알림 대상이 Query 의 경우



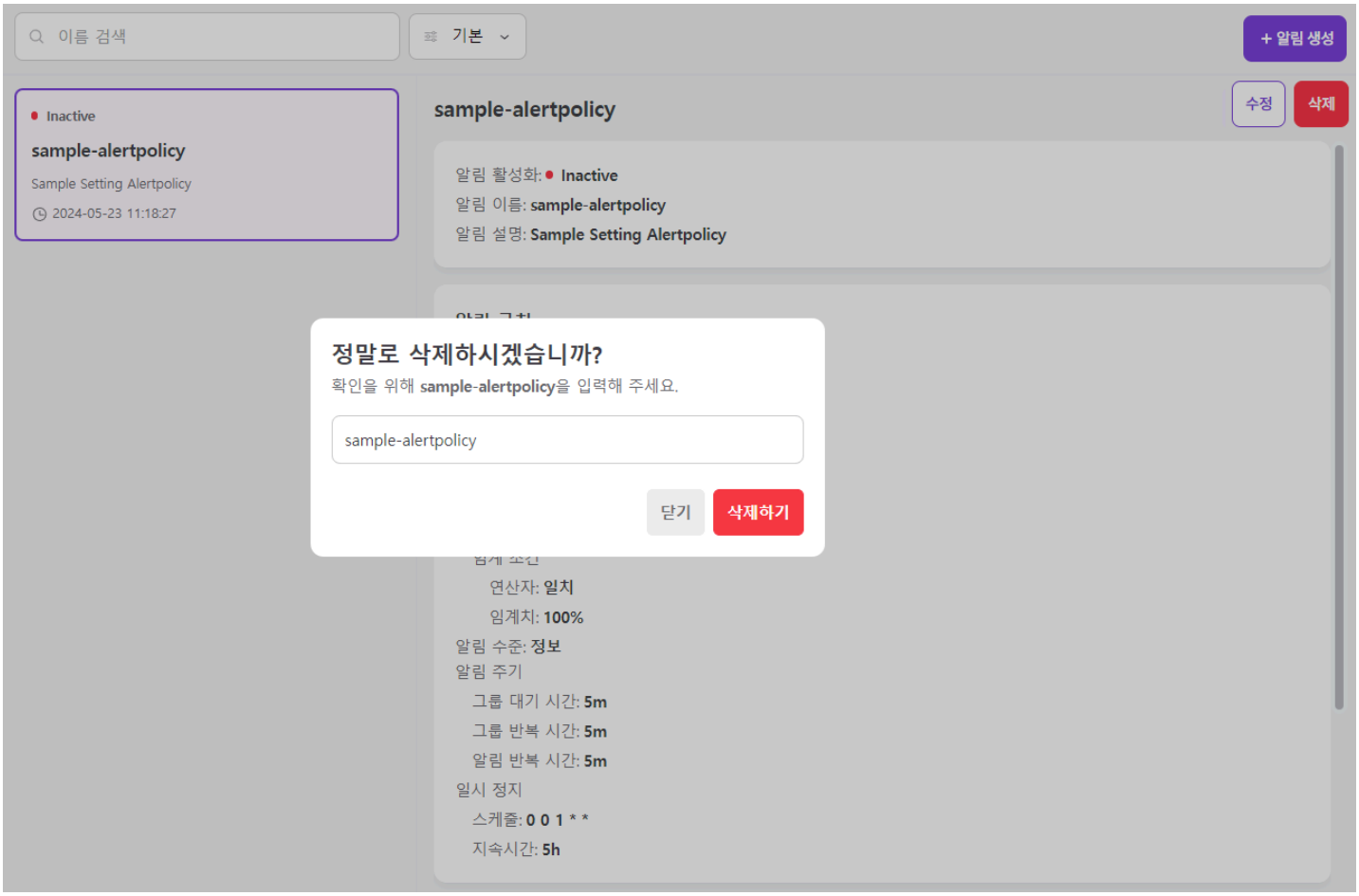

| Query | Prometheus Query 입력 |
| --- | --- |
| 임계 조건 | • 일치<br>• 불일치<br>• 보다 크다<br>• 크거나 같다<br>• 보다 작다<br>• 작거나 같다 |
| | 임계 조건의 값에 해당하는 query 혹은 숫자 입력 |

### 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

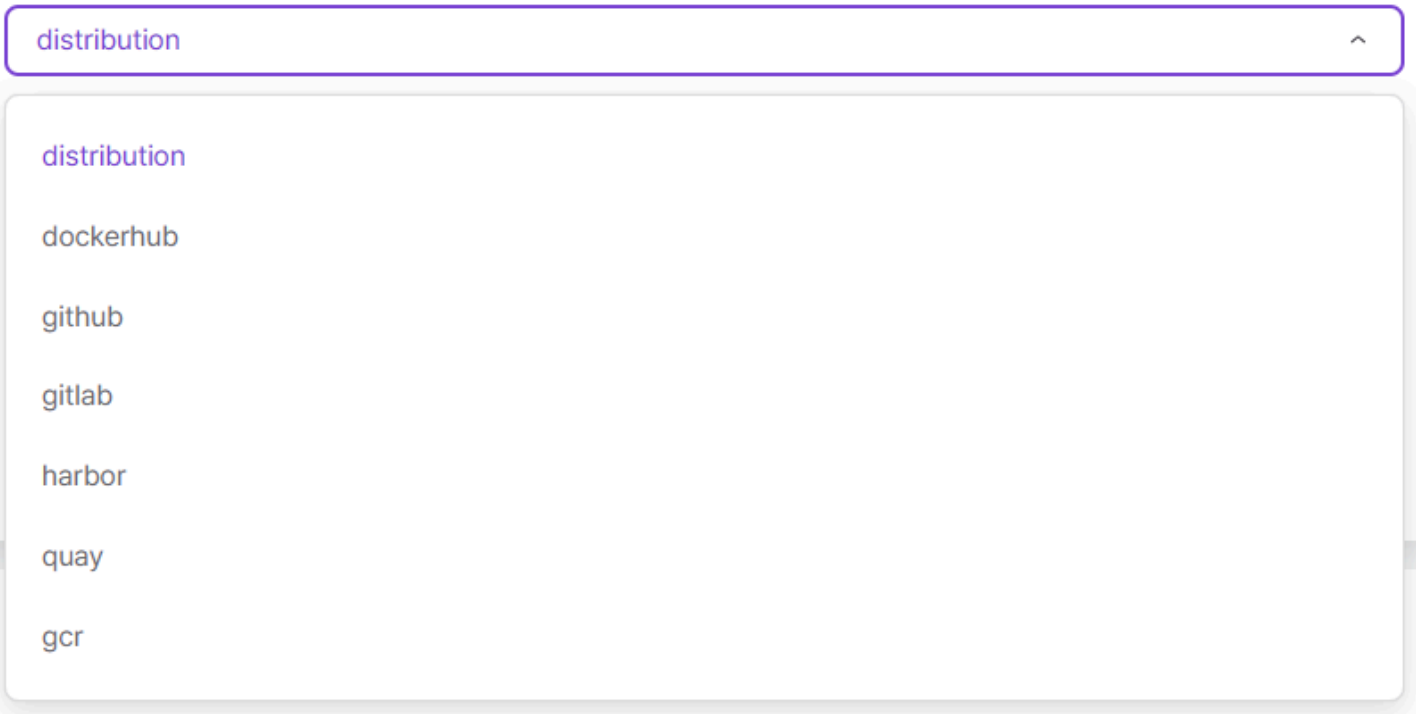
레지스트리는 클러스터에서 공통으로 사용할 컨테이너 이미지 저장소를 관리한다.
이 저장소는 클러스터에 배포된 네임스페이스에서 접근할 수 있다.
아코디언에서는 기본으로 인프라 레지스트리와 사용자 레지스트리를 제공한다.

**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

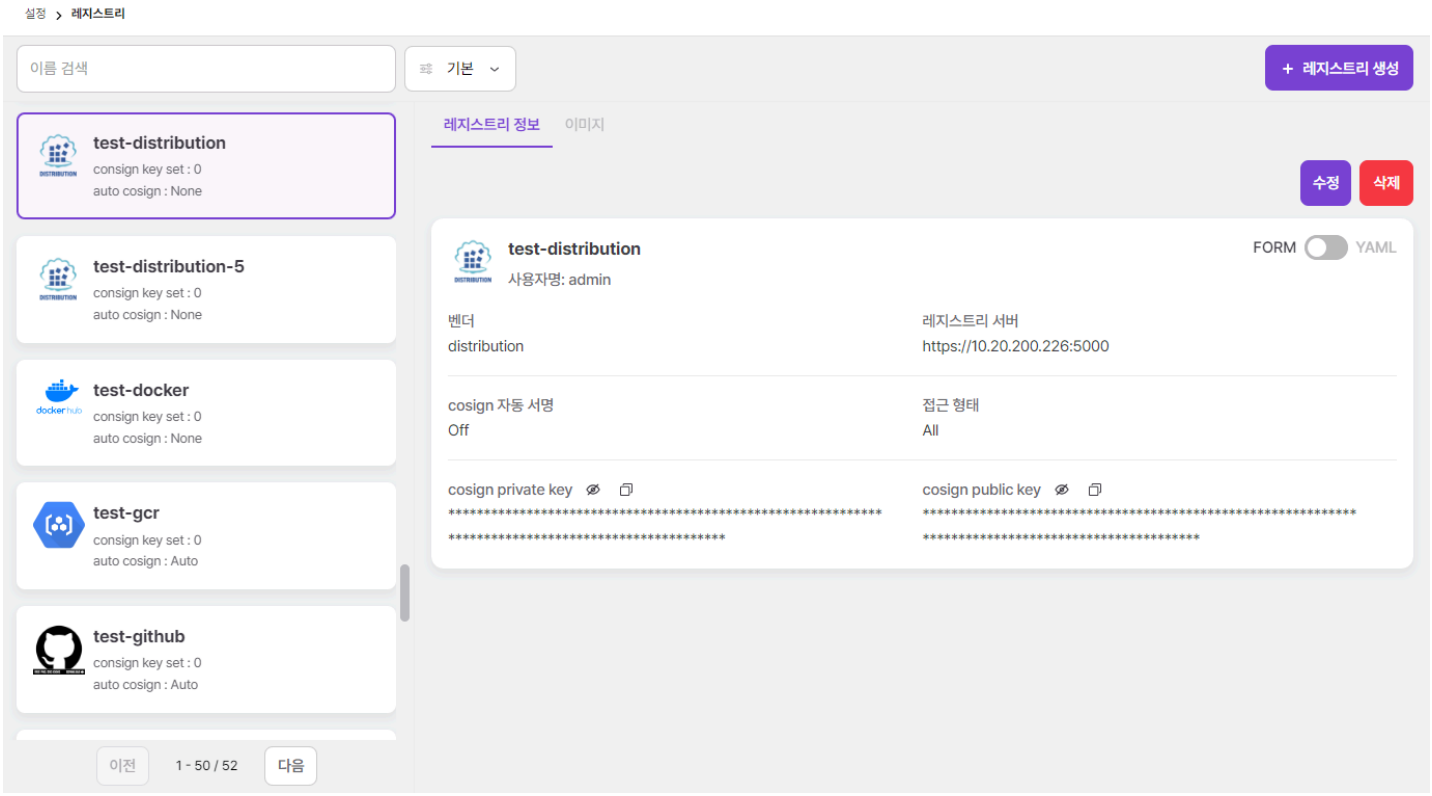



| 항목 | (이미지, 태그)목록 조회 | 상세 조회 | Cosign 서명 | 삭제 |
|---|---|---|---|---|
| distribution | O | O | O | O |
| dockerhub | O | O | X | X |
| github | O | O | O | X |
| gitlab | O | O | O | O |
| harbor | O | O | O | O |
| quay | O | O | O | O |

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

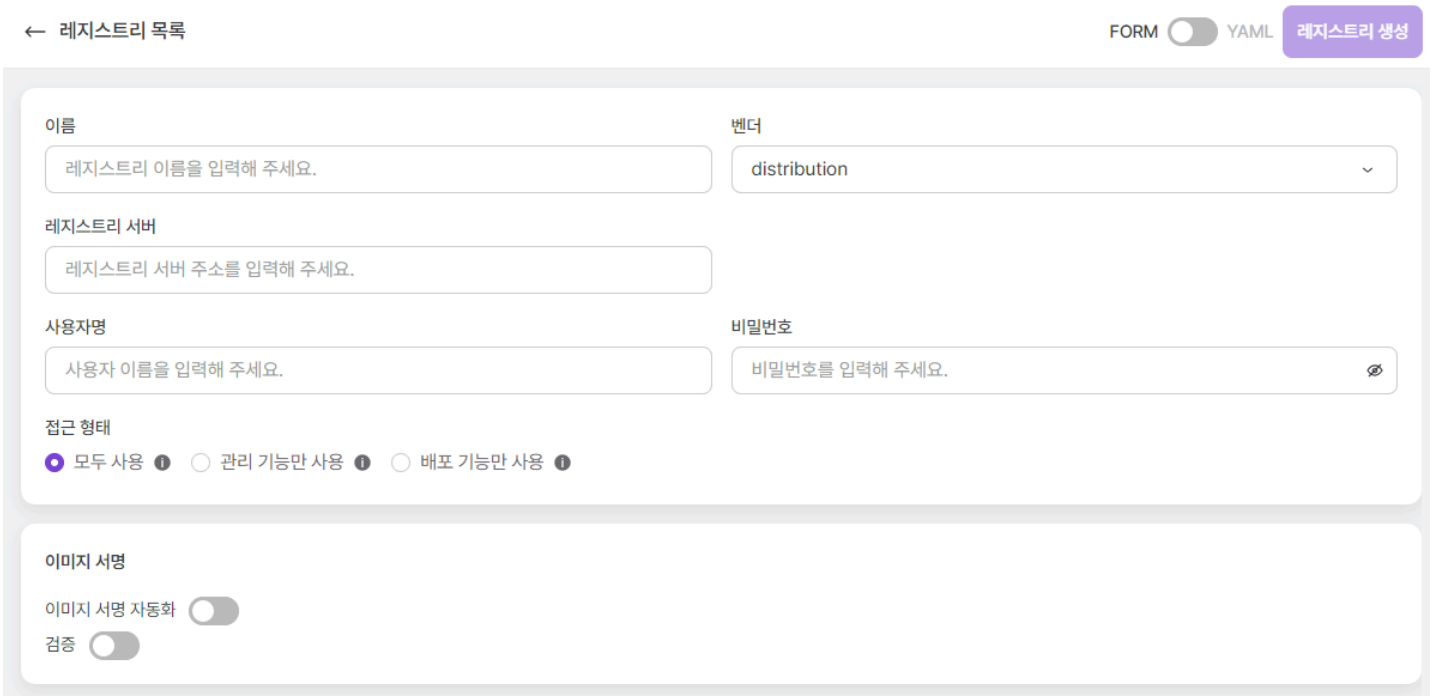
## 레지스트리 목록

레지스트리 메뉴 좌측에는 생성한 레지스트리들의 목록을 표시한다.
카드 형식으로 레지스트리의 이름과 코사인 자동서명 여부 및 추가로 등록한 검증 키의 개수를 확인할 수 있다.



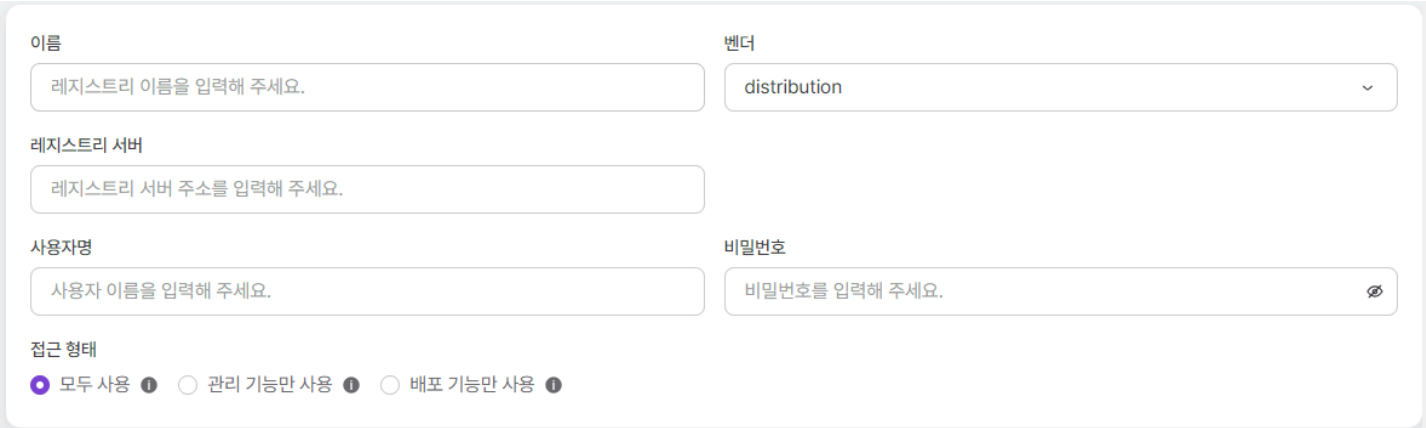

## 레지스트리 생성

+ 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



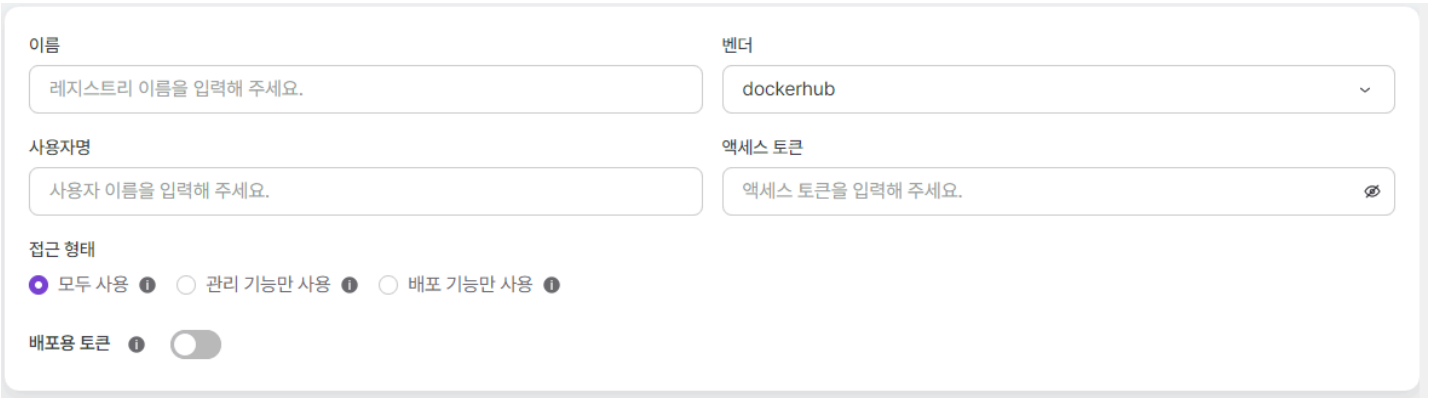

| 항목 | 설명 |
| --- | --- |
| 이름 | 레지스트리 이름 (필수) |
| 레지스트리 서버 | 레지스트리 서버 주소로 Http 와 Https를 구분한다. (필수) |
| 사용자명 | 레지스트리 사용자 이름 (필수) |
| 비밀번호 | 비밀번호 (필수) |

**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



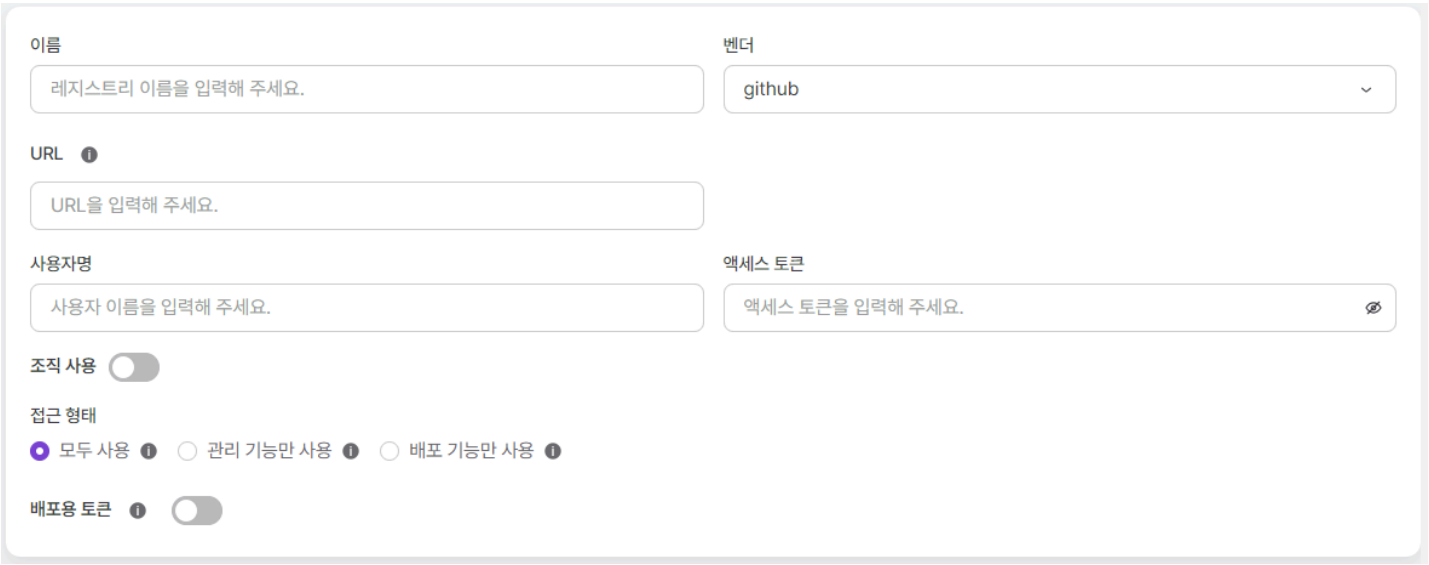

| 항목 | 설명 |
| --- | --- |
| 이름 | 레지스트리 이름 (필수) |
| 사용자명 | 레지스트리 사용자 이름 (필수) |
| 액세스 토큰 | DockerHub에서 발급한 개인 액세스 토큰 (필수) |

**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)을 확인한다.

GitHub FORM




| 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. |

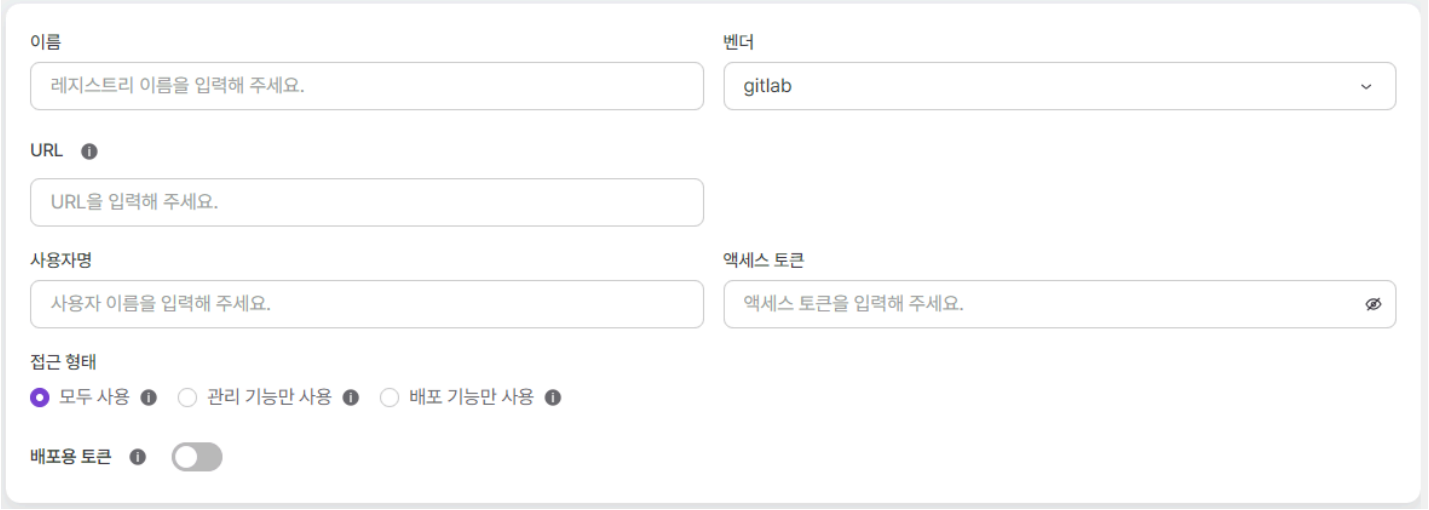
**TIP**

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




| 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**

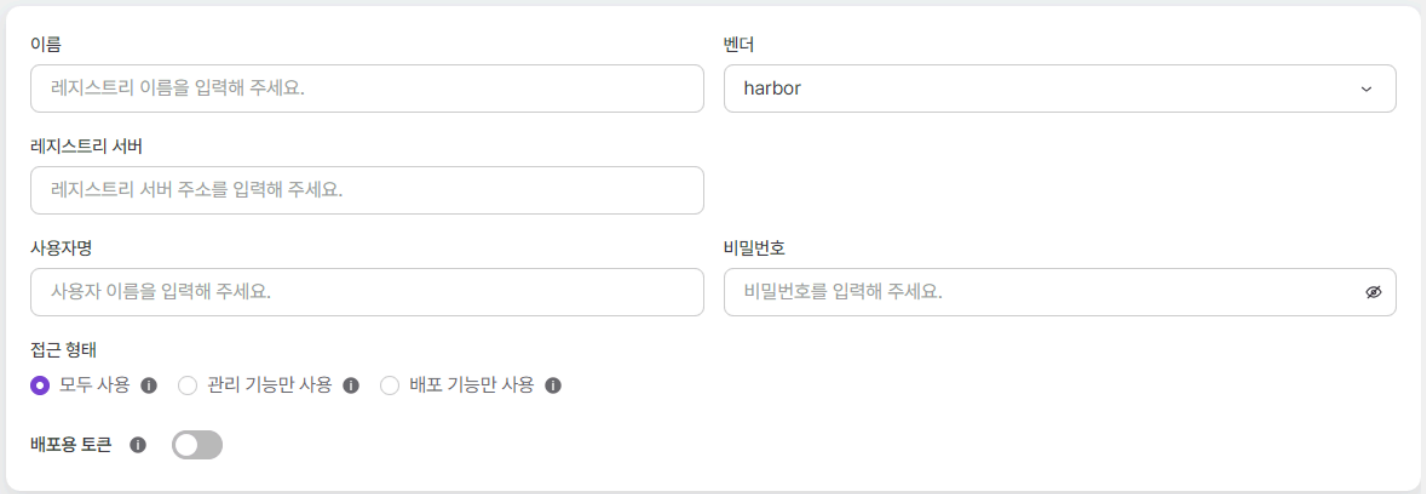
URL 입력은 (사용자명 or 조직명)/(프로젝트명)으로 입력한다.

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)을 확인한다.

Harbor FORM



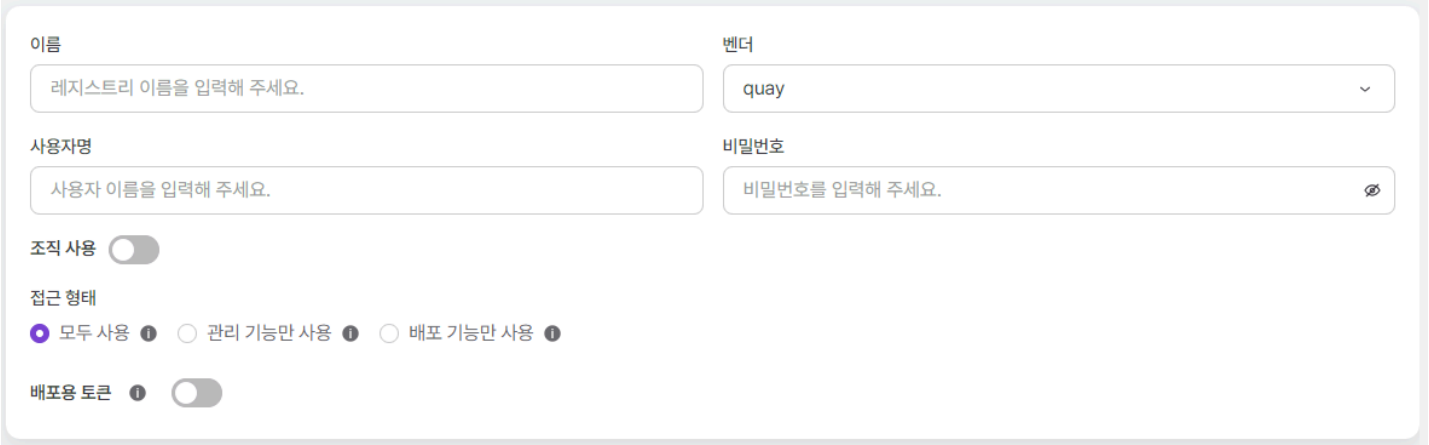

| 항목 | 설명 |
| --- | --- |
| 이름 | 레지스트리 이름 (필수) |
| 레지스트리 서버 | 레지스트리 서버 주소로 Http 와 Https를 구분한다. (필수) |
| 사용자명 | 레지스트리 사용자 이름 (필수) |
| 비밀번호 | 비밀번호 (필수) |

**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

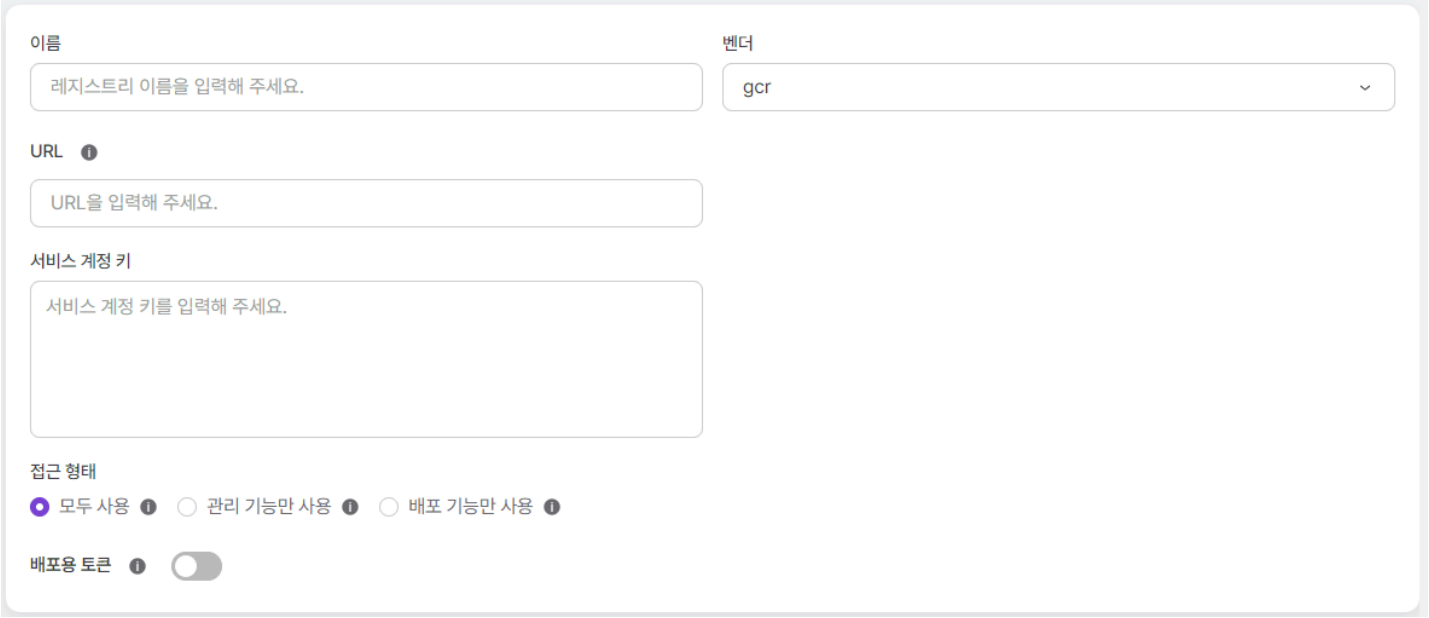



| 항목 | 설명 |
| --- | --- |
| 이름 | 레지스트리 이름 (필수) |
| 사용자명 | 레지스트리 사용자 이름 (필수) |
| 비밀번호 | Quay에서 설정한 도커 로그인 비밀번호 (필수) |
| 조직 사용 | 만약 레지스트리가 조직을 사용할 경우 체크한 뒤 조직 이름을 입력한다. |

**TIP**

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

Gcr FORM




| 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.

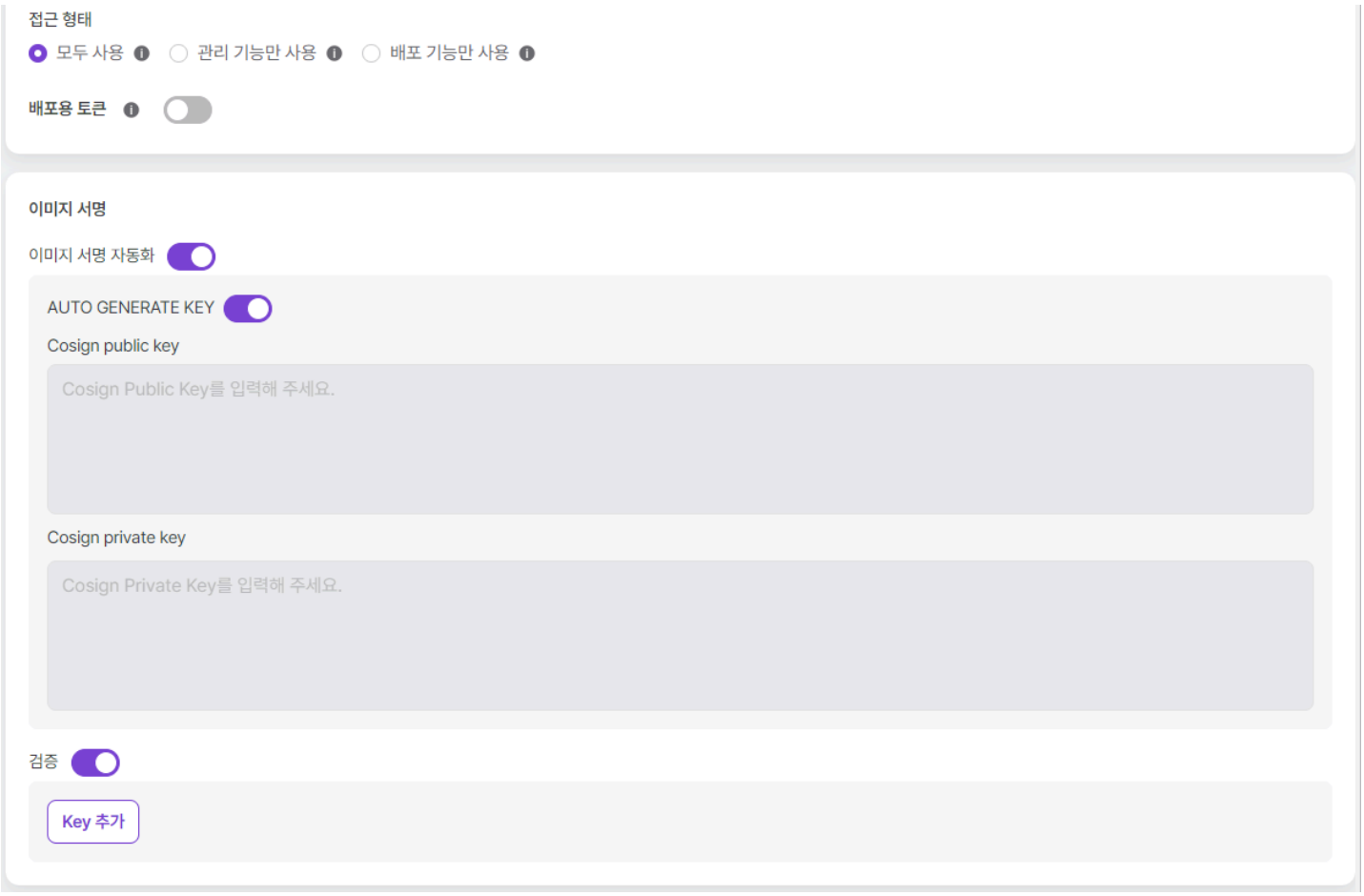
서비스 계정 권한에 대한 설정 방법은 다음을 확인한다.

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

다음은 공통적으로 존재하는 FORM이다.



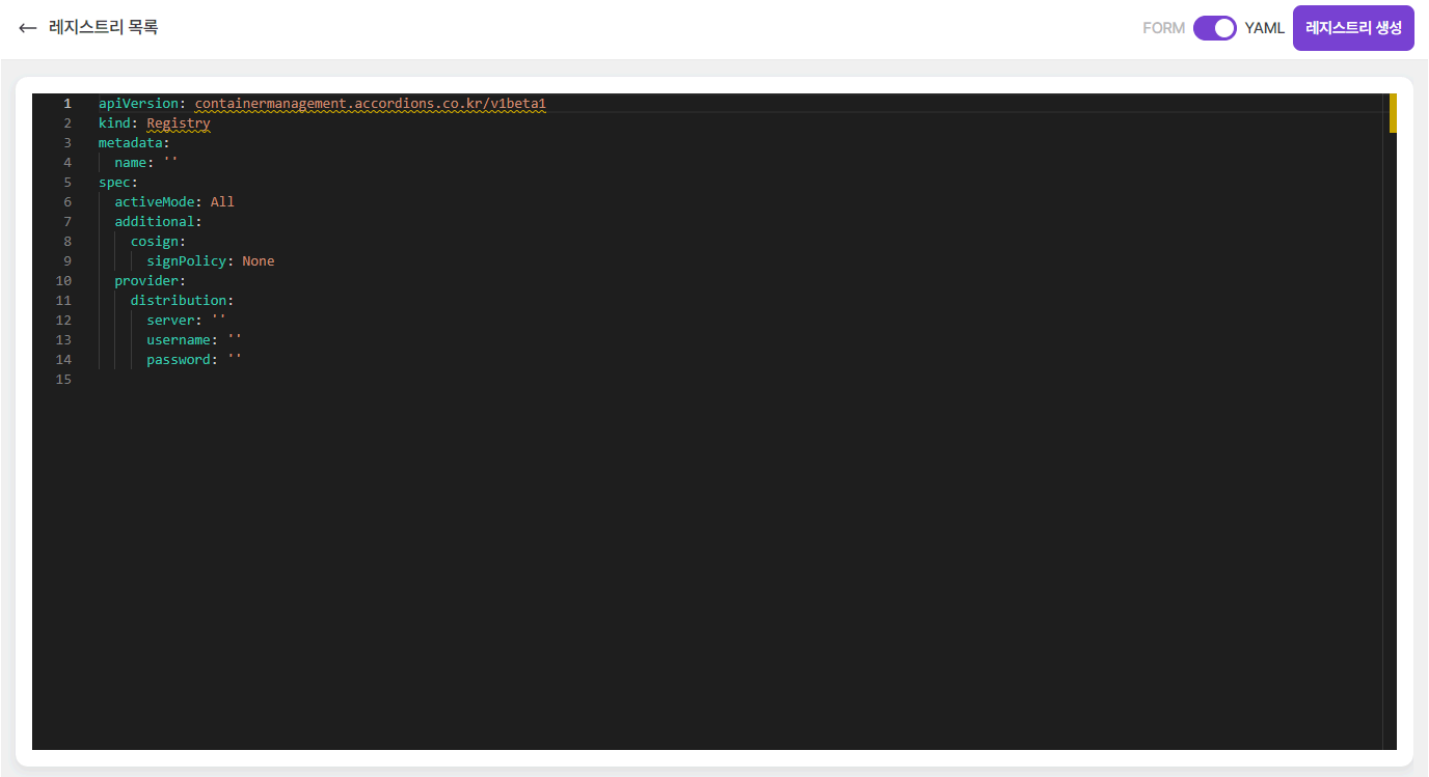

| 항목 | 설명 |
| --- | --- |
| 접근 형태 | 레지스트리 이용 용도에 따라 사용할 모드를 선택한다.<br>• 모두 사용 : 관리와 배포 기능을 모두 사용한다.<br>• 관리 기능만 사용 : 컨테이너 이미지 저장소 내, 이미지 조회나 삭제와 같은 이미지 관리 기능만 사용한다.<br>• 배포 기능만 사용 : 컨테이너 이미지 저장소 인증 정보 시크릿을 각 네임스페이스마다 배포하여 파이프라인/카탈로그에서 사용한다. |
| 배포용 토큰 | 접근 형태 모두 사용 시, 관리와 배포에 대한 인증 권한 정보를 분리하고 싶을 경우 사용한다. 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, 등록한 키로 서명 검증을 할 수 있다. |

### 레지스트리 생성 (YAML)



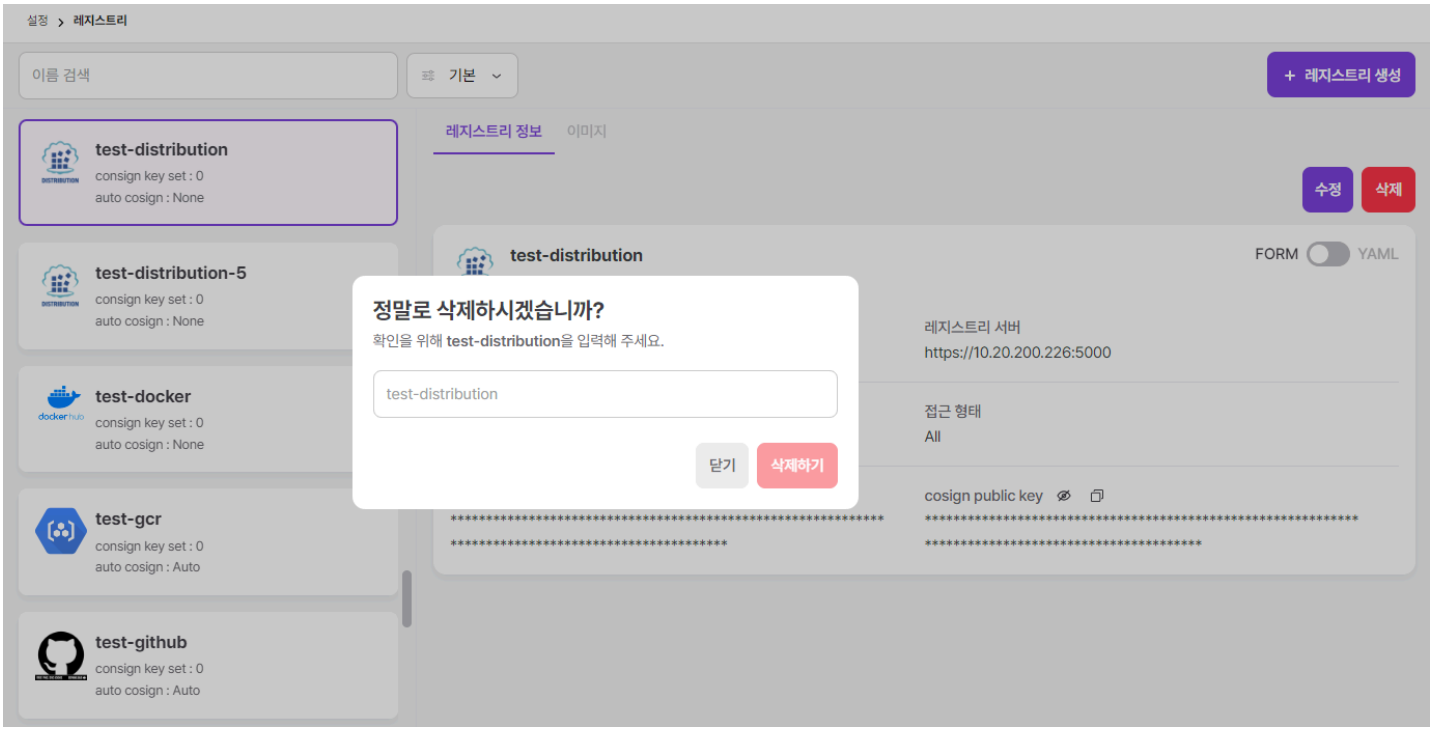

YAML 편집기로도 레지스트리 생성이 가능하다. YAML 편집기로 생성할 경우, 벤더의 스펙은 모두 base64 인코딩한 상태로 입력해야한다.

### 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

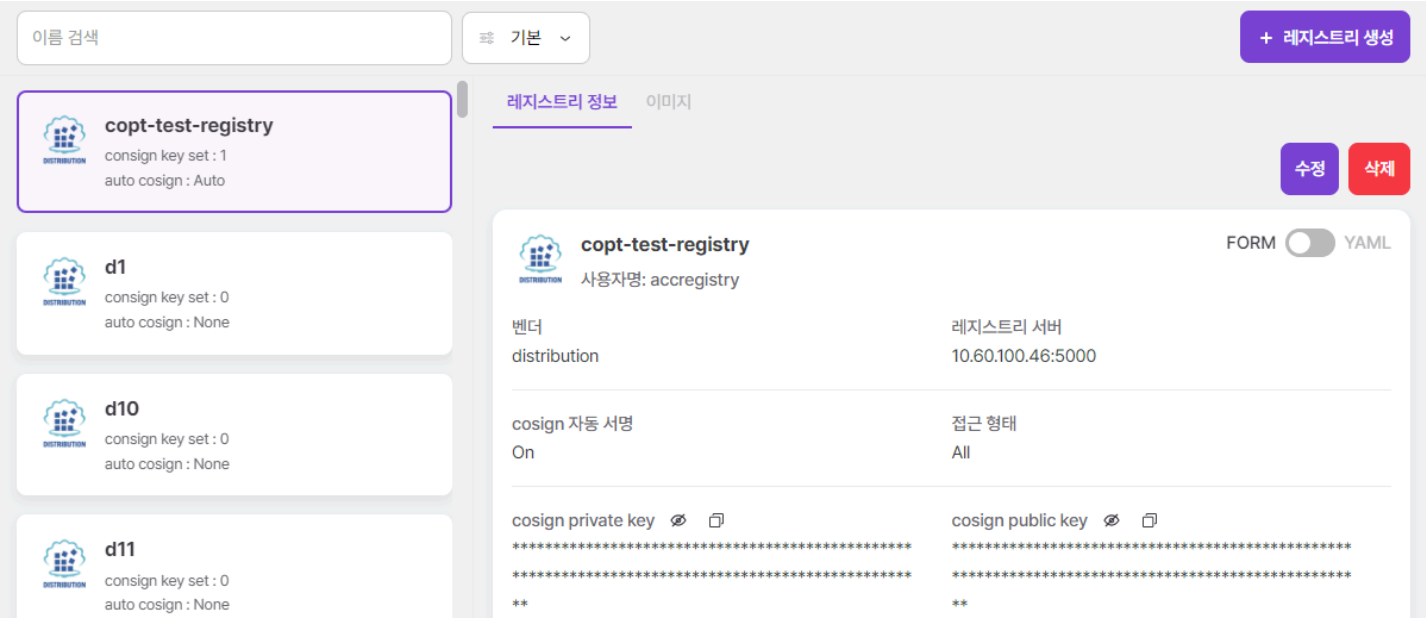
삭제하려는 레지스트리를 선택하고 우측의 `삭제` 버튼을 선택한다.




모달에서 레지스트리 이름을 입력하여 삭제한다.

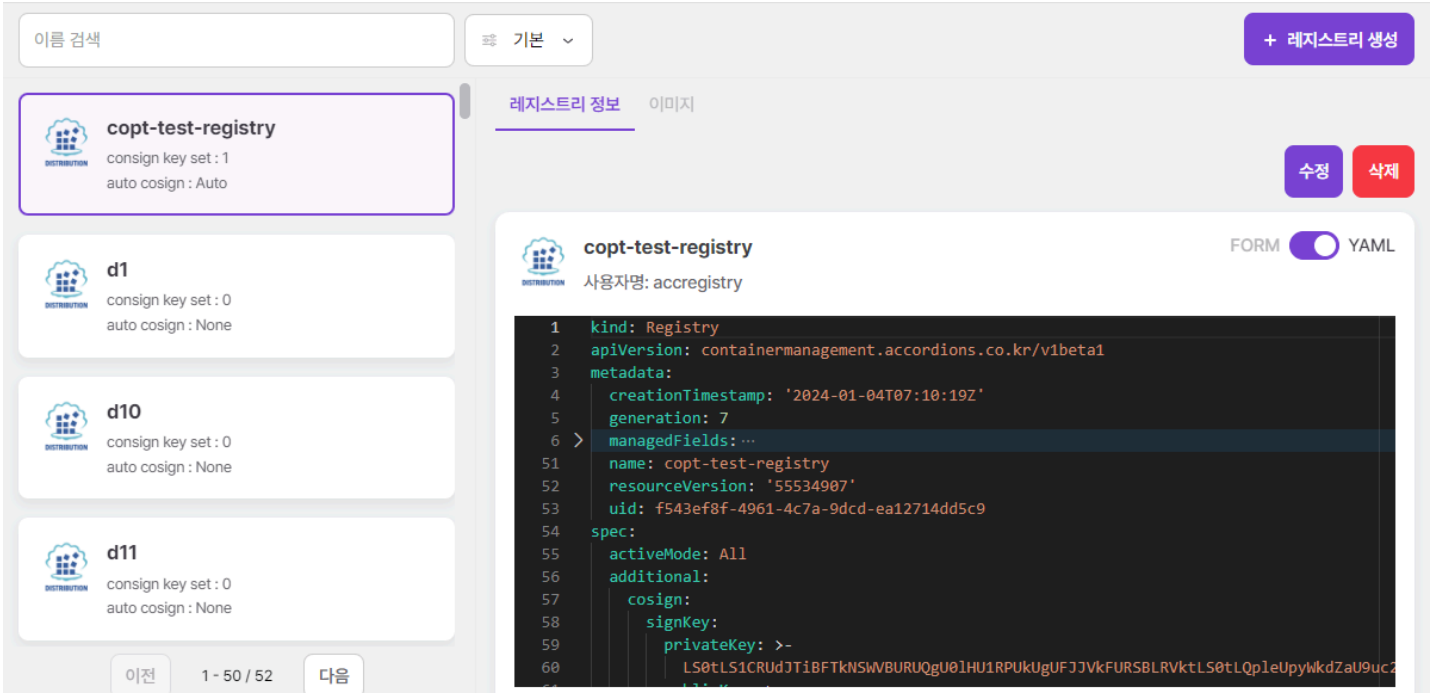
## 레지스트리 정보

레지스트리에 대한 설정 정보를 확인할 수 있다.



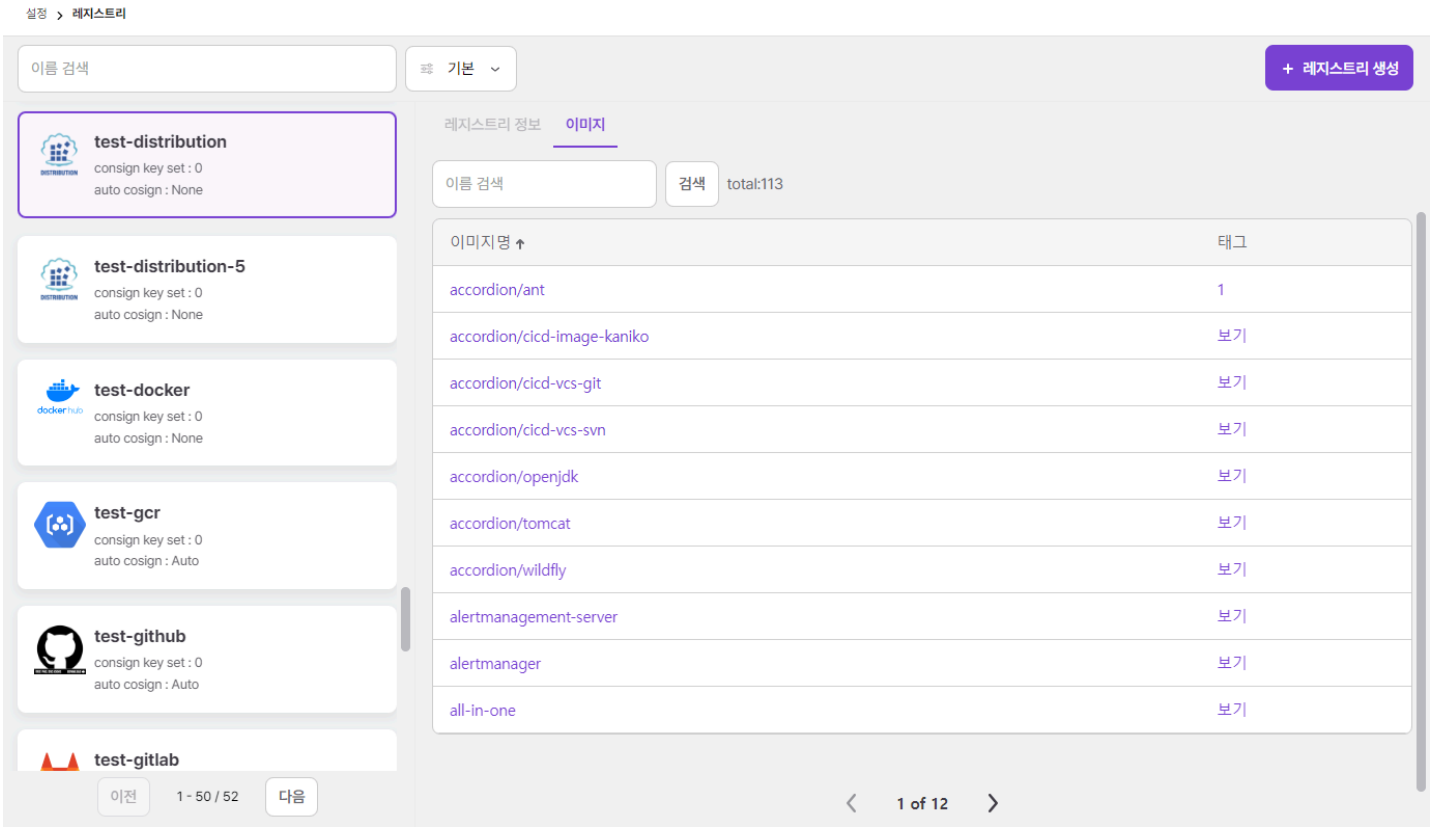

| 항목 | 설명 |
|---|---|
| 이름 | 레지스트리의 이름이 표시된다. |
| 사용자명 | 레지스트리 인증 정보의 사용자명이 표시된다. |
| 벤더 | 레지스트리 벤더 종류가 표시된다. |
| 레지스트리 서버 | 레지스트리 서버가 표시된다. 일부 벤더의 경우 조회 및 배포에 필요한 리포지터리 스코프까지 표시된다. |
| cosign 자동 서명 | Cosign 자동 서명 여부가 표시된다. |
| 접근 형태 | 레지스트리 사용 모드가 표시된다. |
| cosign private key | Cosign 서명에 사용하는 private 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
kind: Registry
apiVersion: containermanagement.accordions.co.kr/v1beta1
metadata:
creationTimestamp: '2024-01-04T07:10:19Z'
generation: 7
managedFields: ···
name: copt-test-registry
resourceVersion: '55534907'
uid: f543ef8f-4961-4c7a-9dcd-ea12714dd5c9
spec:
activeMode: All
additional:
cosign:
signKey:
privateKey: >-
LS0tLS1CRUdJTiBFTkNSWVBURUQgU0lHU1RPUkUgUFJJVkFURSBLRVktLS0tLQpleUpyWkdZaU9uc2

## 이미지 (리포지터리) 목록

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

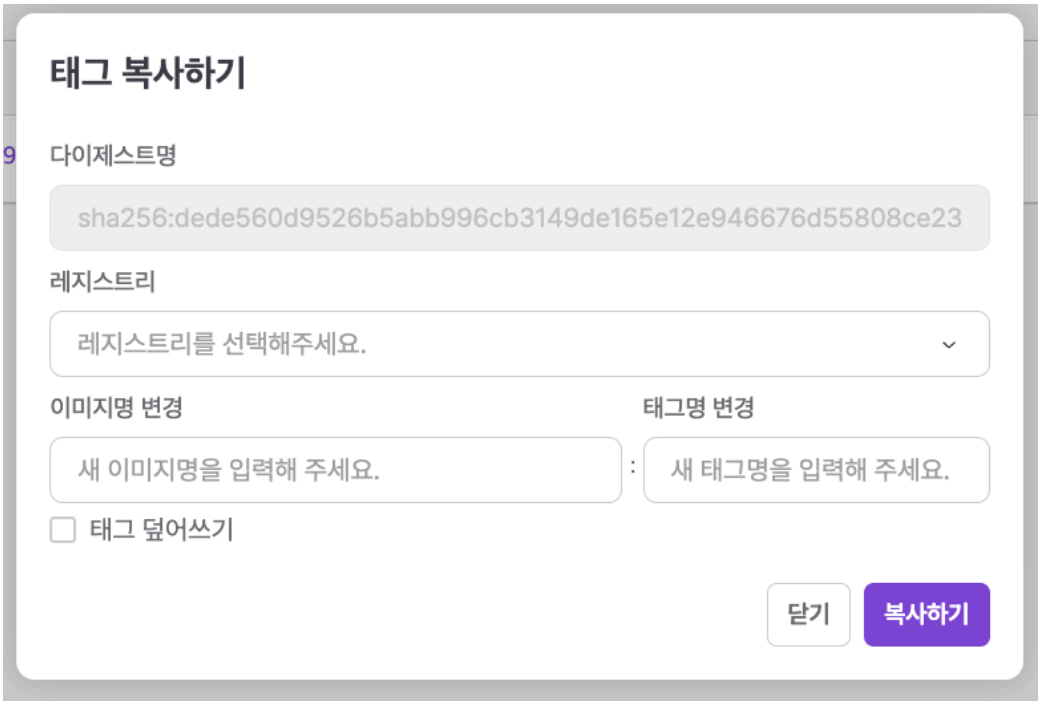
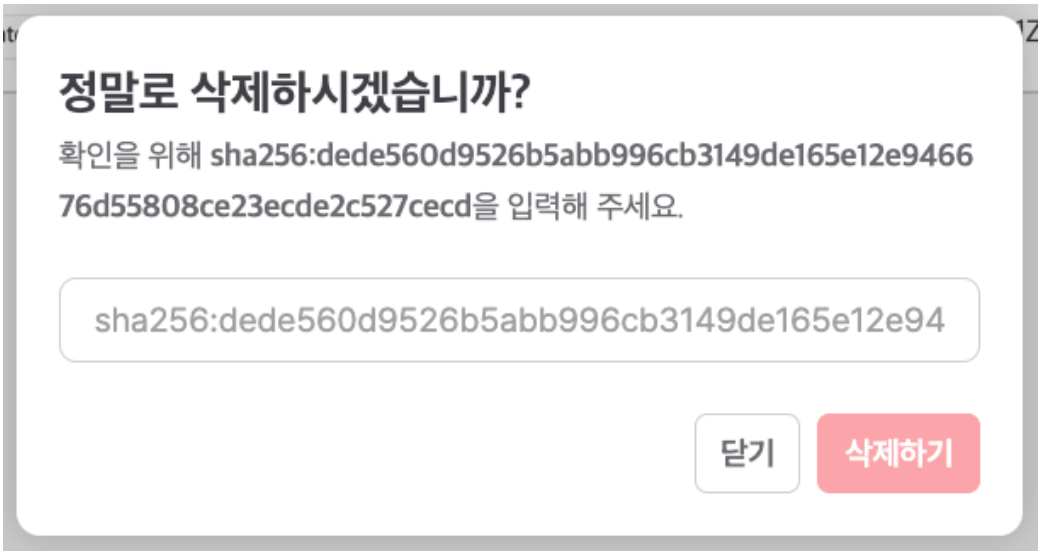



| 항목 | 설명 |
| --- | --- |
| total | 조회한 레지스트리의 총 이미지 수가 표시된다. |
| 이미지명 | 레지스트리의 이미지 저장소 이름이 표시된다. |
| 태그 | 보기 버튼을 선택하면 레지스트리 이미지 저장소가 가지고 있는 총 태그의 수가 표시된다. |

## 이미지 다이제스트 목록

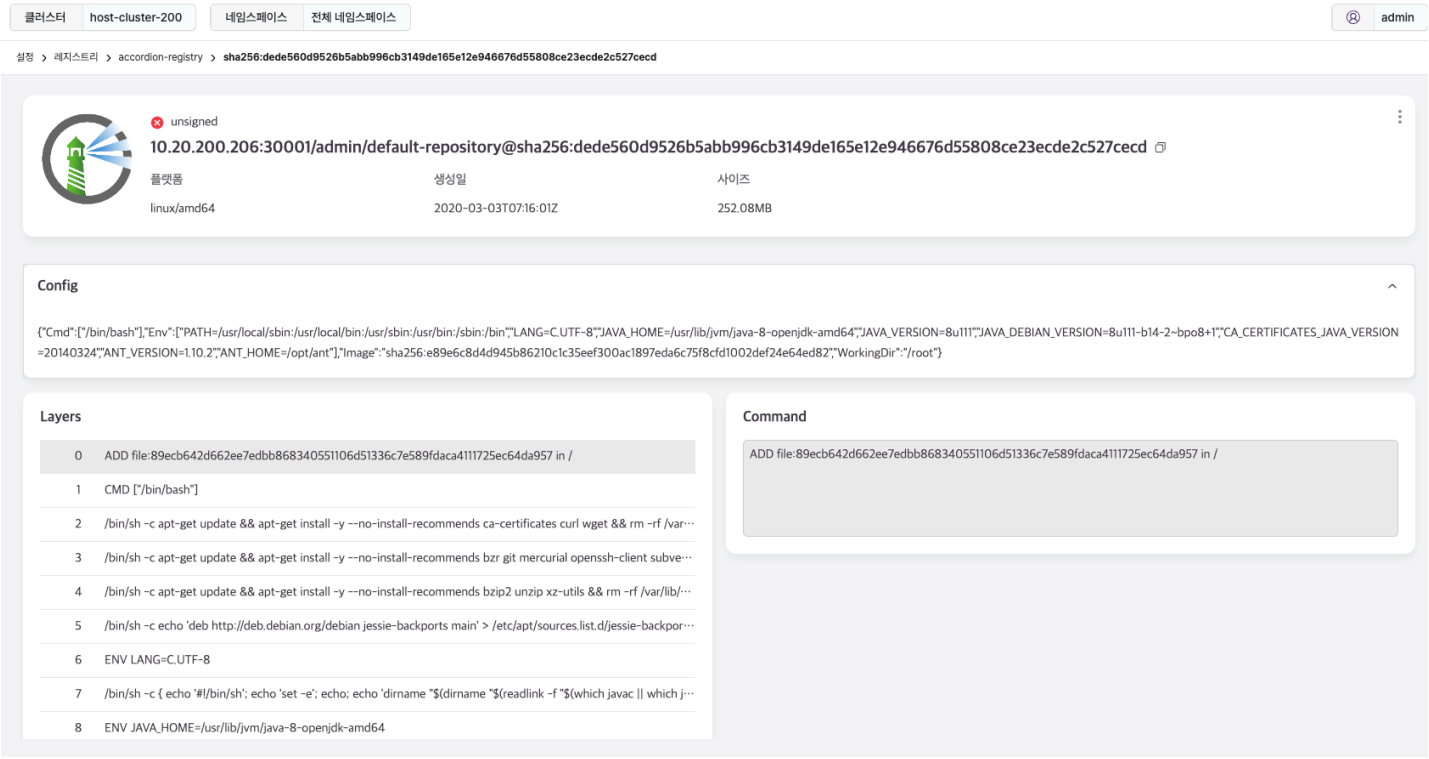
리포지터리에서 사용되는 다이제스트를 목록을 보여준다.
`sig 태그 감추기` 옵션을 통해 순수한 이미지 다이제스트만 목록을 호출할 수 있다.
(`sig 태그` 란 Cosign 시 생성되는 태그다.)

| 항목 | 설명 |
| --- | --- |
| 다이제스트 | 다이제스트명 |
| 태그 | 다이제스트가 가지고 있는 태그의 목록 |
| 서명 여부 | 다이제스트의 Cosign 상태를 나타낸다.<br>signed: 서명된 상태<br>unsigned: 미서명된 상태 |
| 플랫폼 | `manifest` 의 플랫폼 정보가 표시된다.<br>`indexManifests` 의 경우 모든 `manifest` 의 플랫폼 정보가 표시된다. |
| 생성일 | 다이제스트의 생성일.<br>`indexManifests` 의 경우 `N/A` 로 표시된다. |
| 사이즈 | 다이제스트의 사이즈.<br>`indexManifests` 의 경우 `N/A` 로 표시된다. |
| ✍ | <br>DockerHub를 제외한 레지스트리 중 이미지 서명 자동화를 설정한 경우 Cosign을 지원한다.<br>레지스트리가 가지고 있는 privateKey 를 이용하여 이미지 다이제스트를 Cosign 서명한다.<br>만약 indexManifests인 경우 recursive 하게 하위의 다이제스트들도 모두 서명 처리된다.<br>서명 처리된 다이제스트의 경우, 레지스트리가 가지고 있는 publicKey 또는 verifyKeys 를 이용하여 검증할 수 있다. |

| 항목 | 설명 |
| --- | --- |
| |  <br>다이제스트를 복사한다.<br>복사할 대상의 레지스트리를 선택하고 이미지명, 태그명을 변경할 수 있다.<br>태그명을 변경하지 않을 경우 기본적으로 latest 태그로 복사된다.<br>태그 덮어쓰기 를 선택한 경우 기존에 동일한 태그명이 존재할 경우 무시하고 덮어씌워진다. |
| |  <br>DockerHub, GitHub, Gcr을 제외한 벤더인 경우 삭제를 지원한다.<br>삭제하기를 선택할 경우 다이제스트를 삭제한다. |

## 이미지 다이제스트 정보

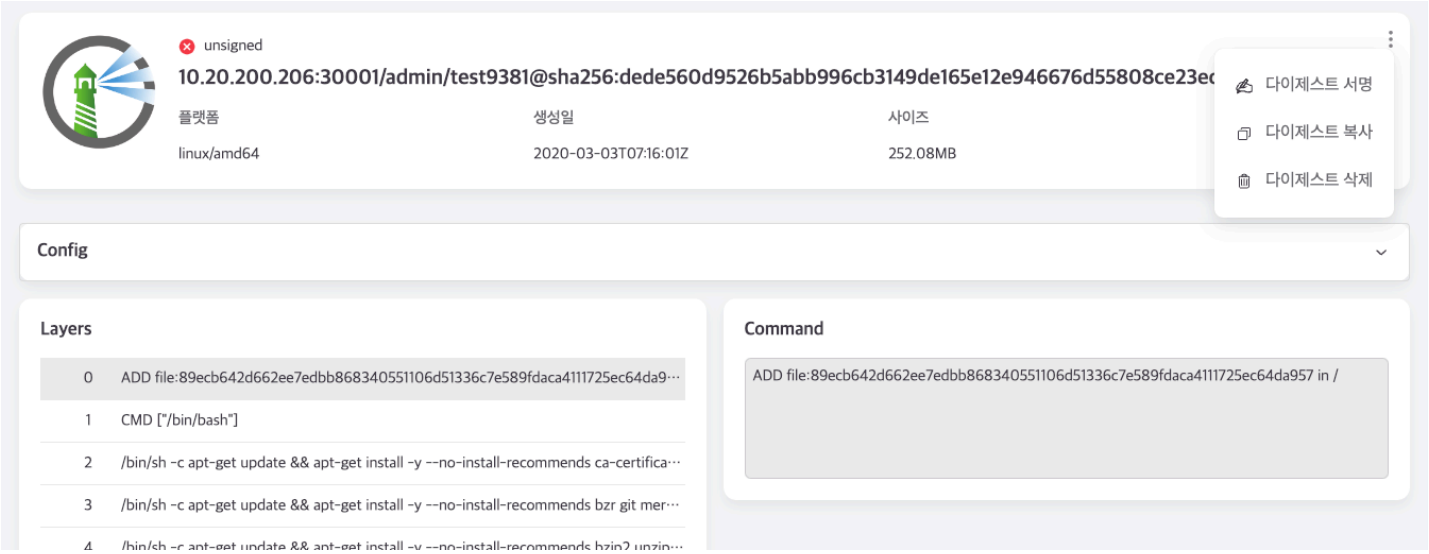
다이제스트 정보를 manifest 기반으로 보여준다.




다이제스트 관련 동작 버튼

우측 상단의 ⋮ 을 누르면 다이제스트에 대한 서명, 복사, 삭제에 대한 동작을 할 수 있는 셀렉트 박스가 나온다.

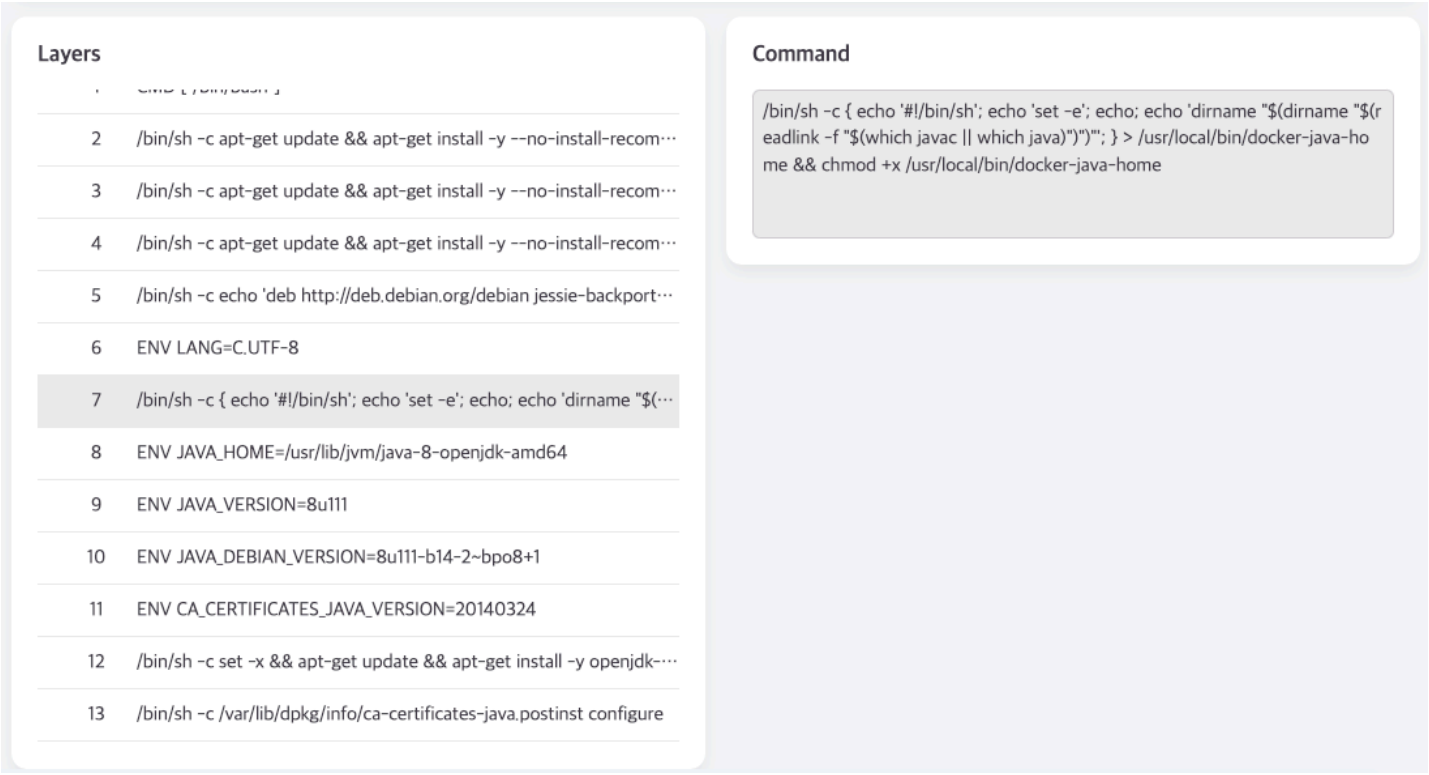
다이제스트 목록의 동작과 동일한 방식으로 동작된다.




| 항목 | 설명 |
| --- | --- |
| 다이제스트 서명 | 다이제스트를 서명한다. |

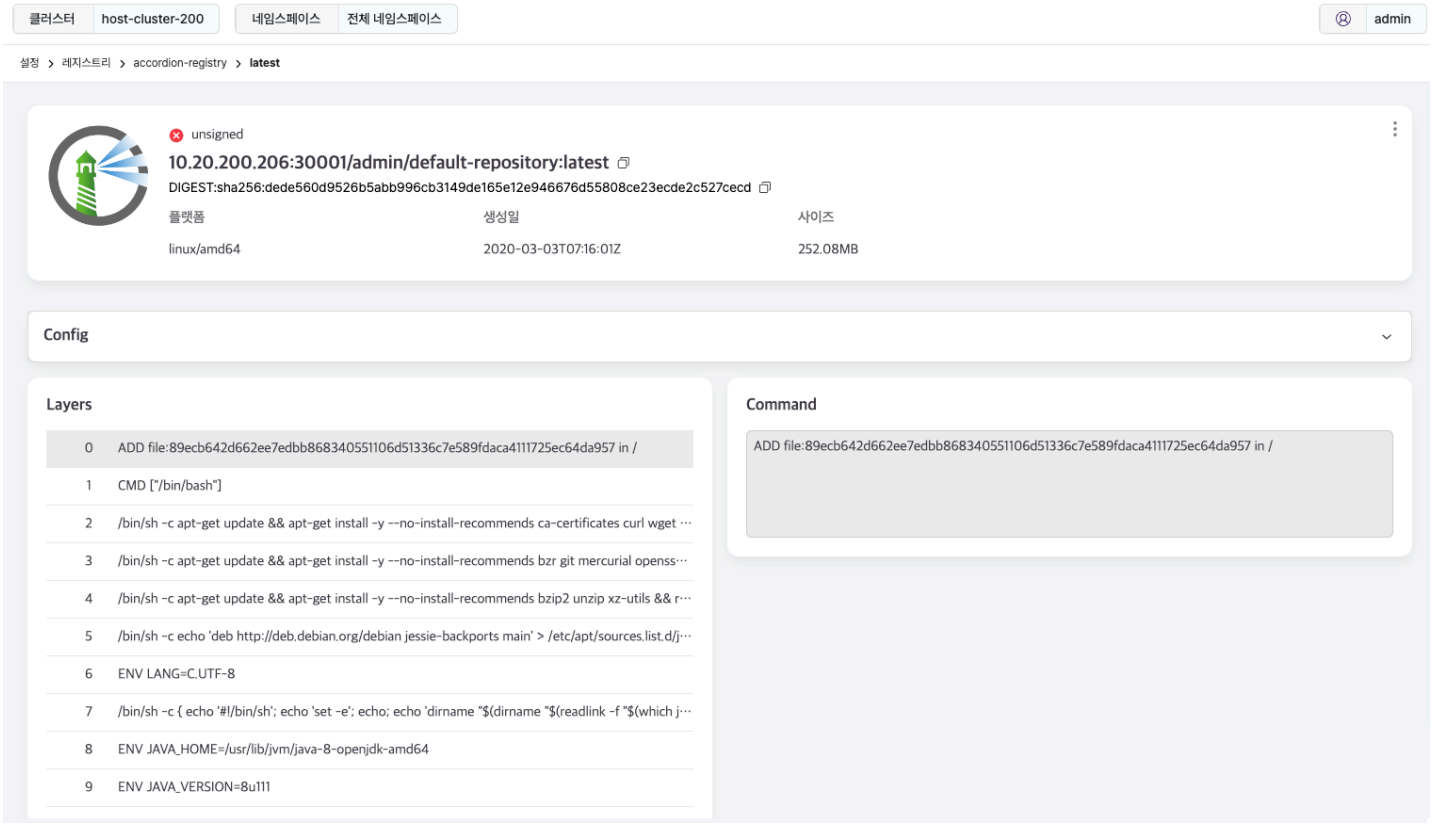
| 항목 | 설명 |
| --- | --- |
| 다이제스트 복사 | 다이제스트를 복사한다.<br>복사할 대상의 레지스트리를 선택하고 이미지명, 태그명을 변경할 수 있다.<br>태그명을 변경하지 않을 경우 기본적으로 `latest` 태그로 복사된다.<br>`태그 덮어쓰기` 를 선택한 경우 기존에 동일한 태그명이 존재할 경우 무시하고 덮어씌워진다. |
| 다이제스트 삭제 | 다이제스트를 삭제한다. |

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




## 리포지터리 태그 정보

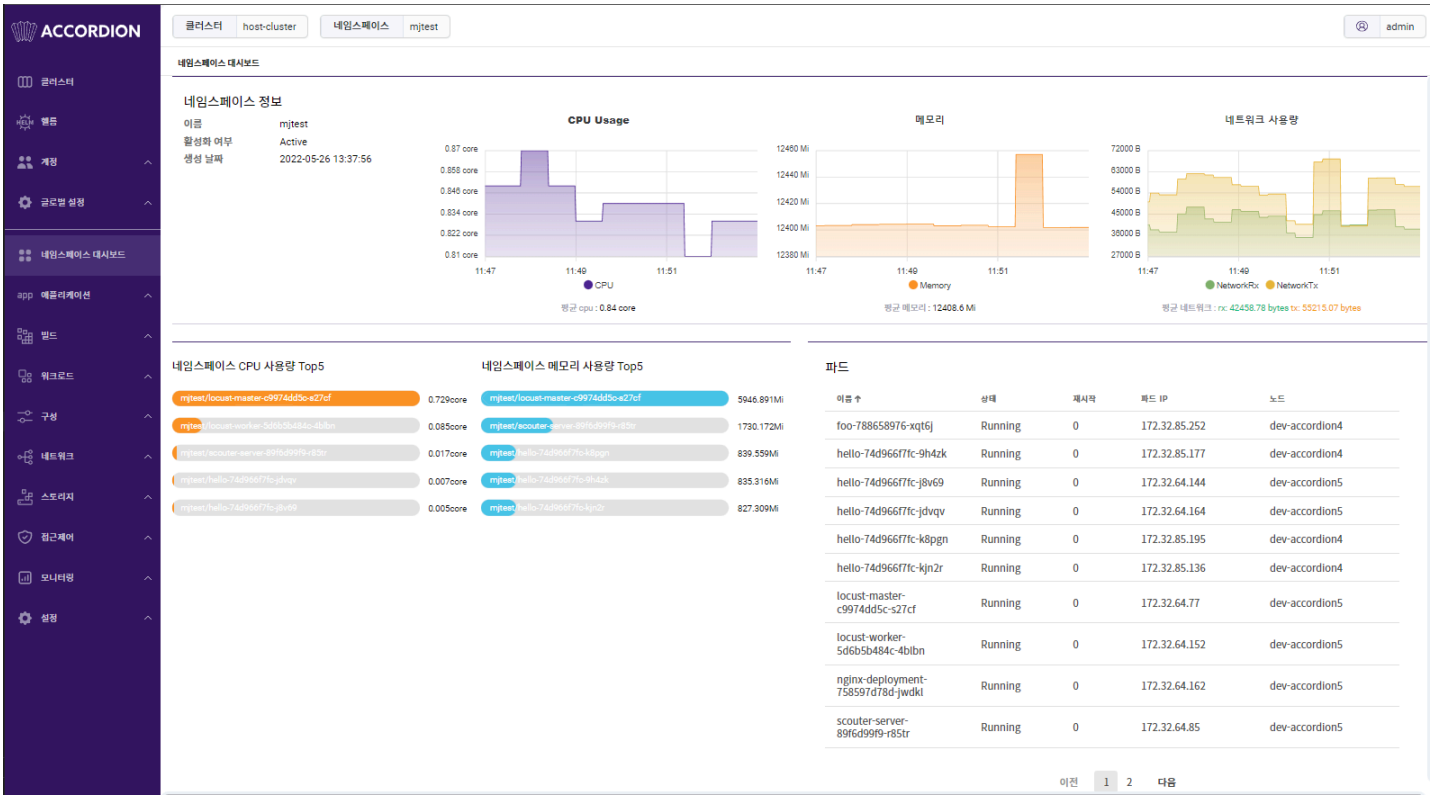
리포지터리 태그 정보는 다이제스트 정보와 거의 동일한 화면을 보여준다.
추가적으로 해당 태그의 `digest` 만 별도로 표시된다.


클러스터
host-cluster-200
네임스페이스
전체 네임스페이스
admin
설정 > 레지스트리 > accordion-registry > latest
unsigned
10.20.200.206:30001/admin/default-repository:latest
DIGEST:sha256:dede560d9526b5abb996cb3149de165e12e946676d55808ce23ecde2c527cecd
플랫폼
생성일
사이즈
linux/amd64
2020-03-03T07:16:01Z
252.08MB
Config
Layers
0 ADD file:89ecb642d662ee7edbb868340551106d51336c7e589fdaca4111725ec64da957 in /
1 CMD ["/bin/bash"]
2 /bin/sh -c apt-get update && apt-get install -y --no-install-recommends ca-certificates curl wget ···
3 /bin/sh -c apt-get update && apt-get install -y --no-install-recommends bzr git mercurial openss···
4 /bin/sh -c apt-get update && apt-get install -y --no-install-recommends bzip2 unzip xz-utils && r···
5 /bin/sh -c echo 'deb http://deb.debian.org/debian jessie-backports main' > /etc/apt/sources.list.d/j···
6 ENV LANG=C.UTF-8
7 /bin/sh -c { echo '#!/bin/sh'; echo 'set -e'; echo; echo 'dirname "$(dirname "$(readlink -f "$(which j···
8 ENV JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
9 ENV JAVA_VERSION=8u111
Command
ADD file:89ecb642d662ee7edbb868340551106d51336c7e589fdaca4111725ec64da957 in /

# 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.

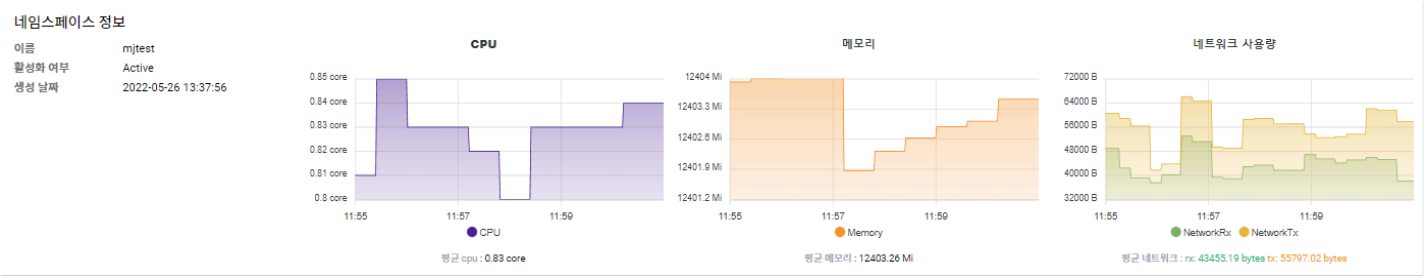
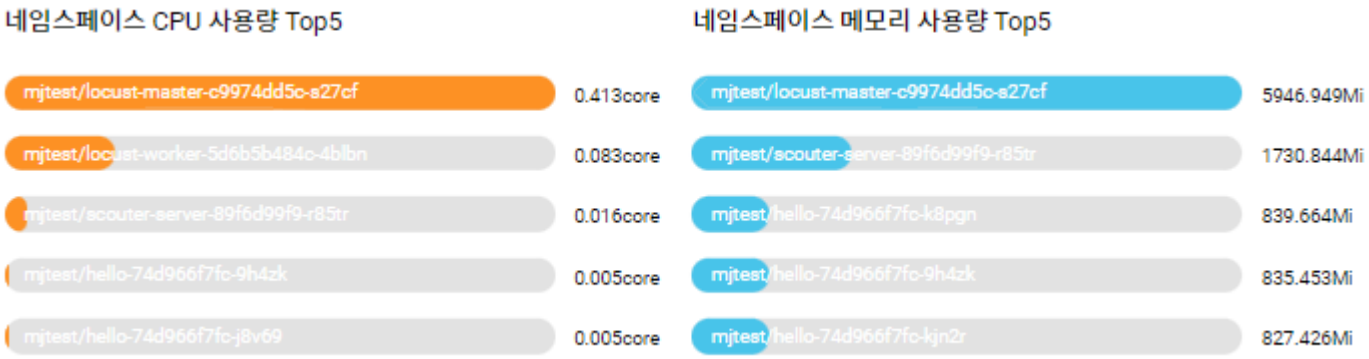



Table 10. 네임스페이스 정보

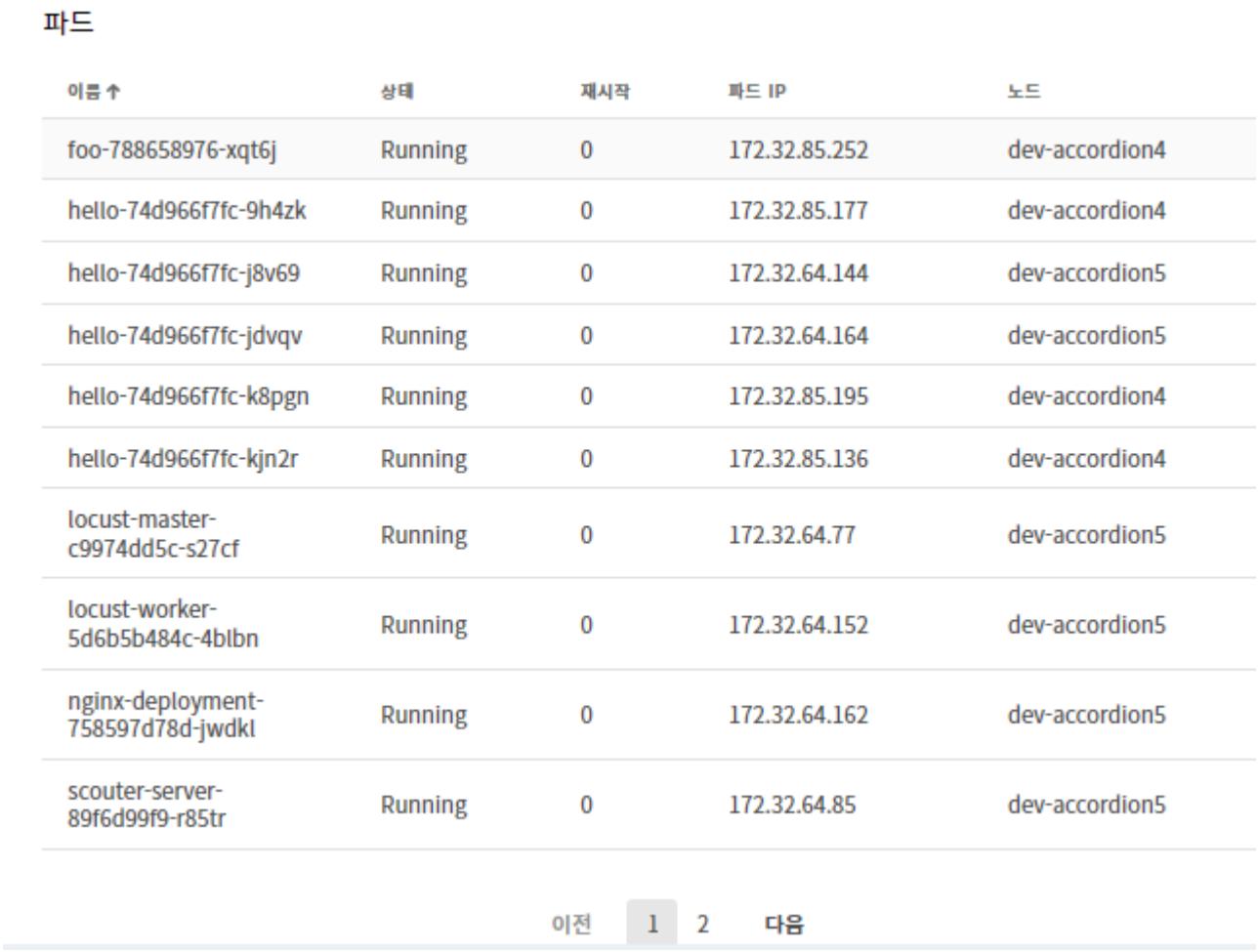
| 항목 | 설명 |
|---|---|
| 이름 | 네임스페이스 이름 |
| 활성화 여부 | 네임스페이스 상태 (Active 또는 Terminating)<br>• Active: 정상 배포<br>• Terminating: 삭제 중 |
| 생성 날짜 | 생성된 시간 |
| CPU | 5분간 CPU 사용량 (평균 CPU: CPU의 평균값) |

| 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.




우측의 목록은 네임스페이스에 배포된 모든 파드의 목록을 나열한다.




| 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 애플레케이션을 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.

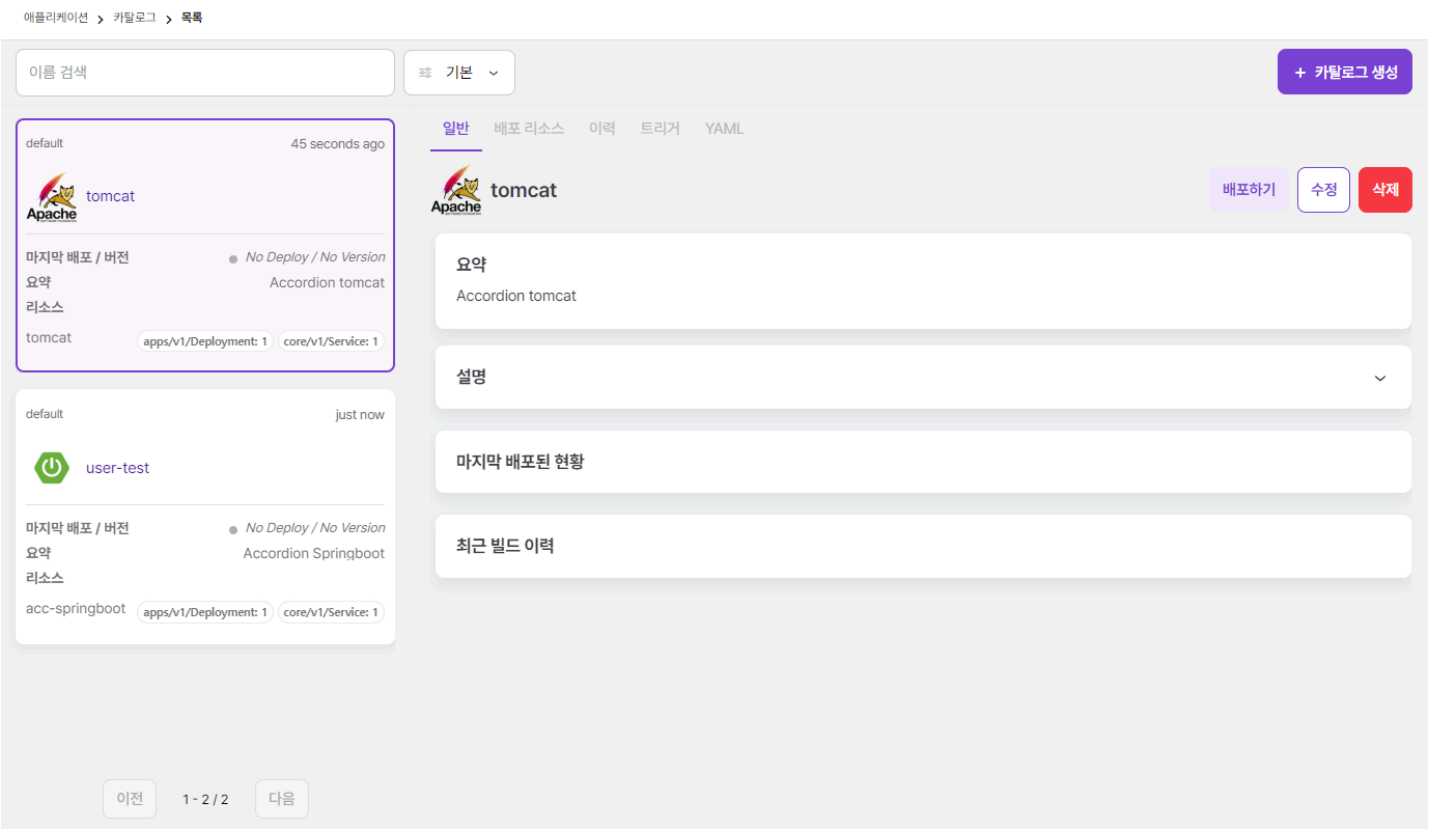
#### 4.3.2.1. 카탈로그

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

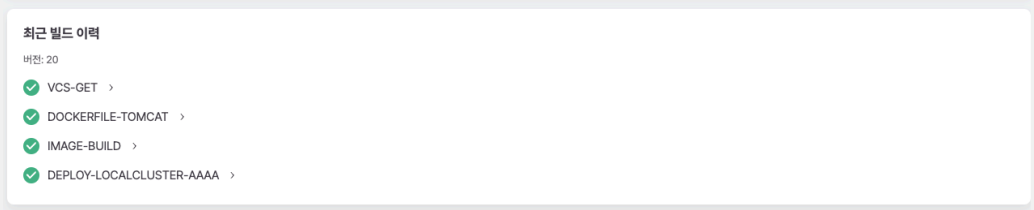
| 특징 | 설명 |
|---|---|
| 다양한 쿠버네티스 리소스 배포 | 디플로이먼트, 스테이트풀셋 등 코어 리소스를 포함하여 커스텀 리소스도 배포할 수 있다. |
| 지속적인 배포 | 카탈로그로 배포한 리소스 집합에 대해 업그레이드 배포 등 라이플사이클을 관리할 수 있다. |
| 배포 이력 관리 | 배포를 수행한 시점의 스펙과 수행을 요청한 사용자 등의 이력을 관리한다. |
| 롤백 | 기존에 성공적으로 배포된 이력에 기술된 배포 리소스 스펙으로 리소스만 다시 생성한다. |
| 다양한 배포 정책 | 단일 카탈로그를 구성하는 리소스들에 대해서도 리소스 각자의 배포 정책을 가져갈 수 있다. |
| 멀티클러스터 배포 지원 | 아코디언에서 관리되는 클러스터에 대해 서로 다른 네임스페이스와 클러스터에 동일한 배포와 라이플사이클을 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.

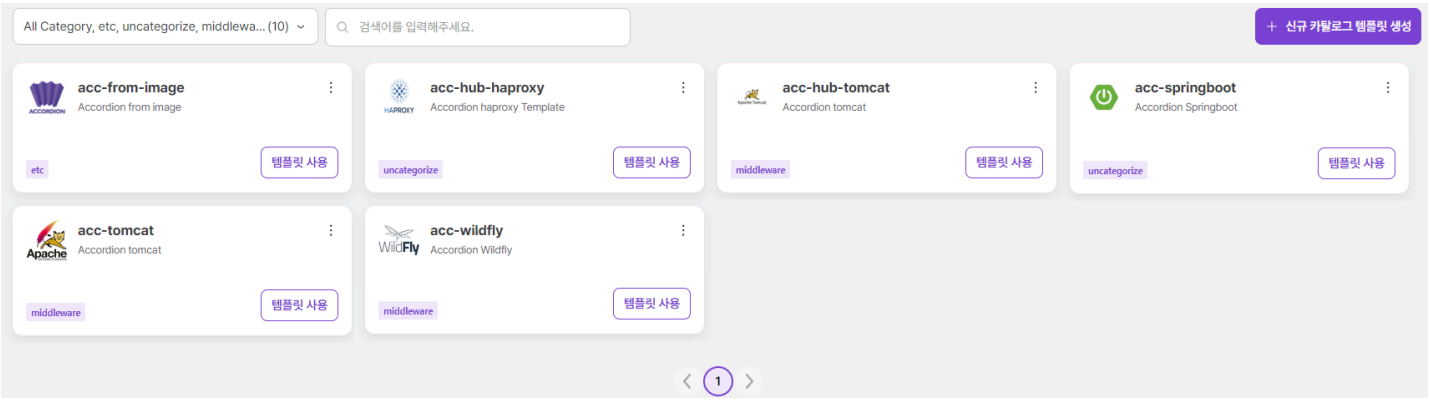


카탈로그 메뉴에서는 카탈로그에 대한 설명과 리소스 구성, 배포 내역을 확인할 수 있다.

| 항목 | 설명 |
| --- | --- |
| 일반 | 카탈로그에 대한 기본 정보와 배포 현황에 대한 정보 제공<br>• 요약<br>• 설명<br>• 마지막 배포된 현황<br>• 최근 빌드 이력<br>  |
| 배포 리소스 | 카탈로그로 배포한 쿠버네티스 리소스 정보 제공 |
| 이력 | 카탈로그로 배포한 이력 정보 제공 및 배포 관리<br>• 카탈로그 배포 과정 조회<br>• 배포 중인 카탈로그의 배포 중단<br>• 배포에 성공한 이력에 대해 롤백 |
| YAML | 카탈로그에 대한 쿠버네티스 리소스 관리 |
| 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.

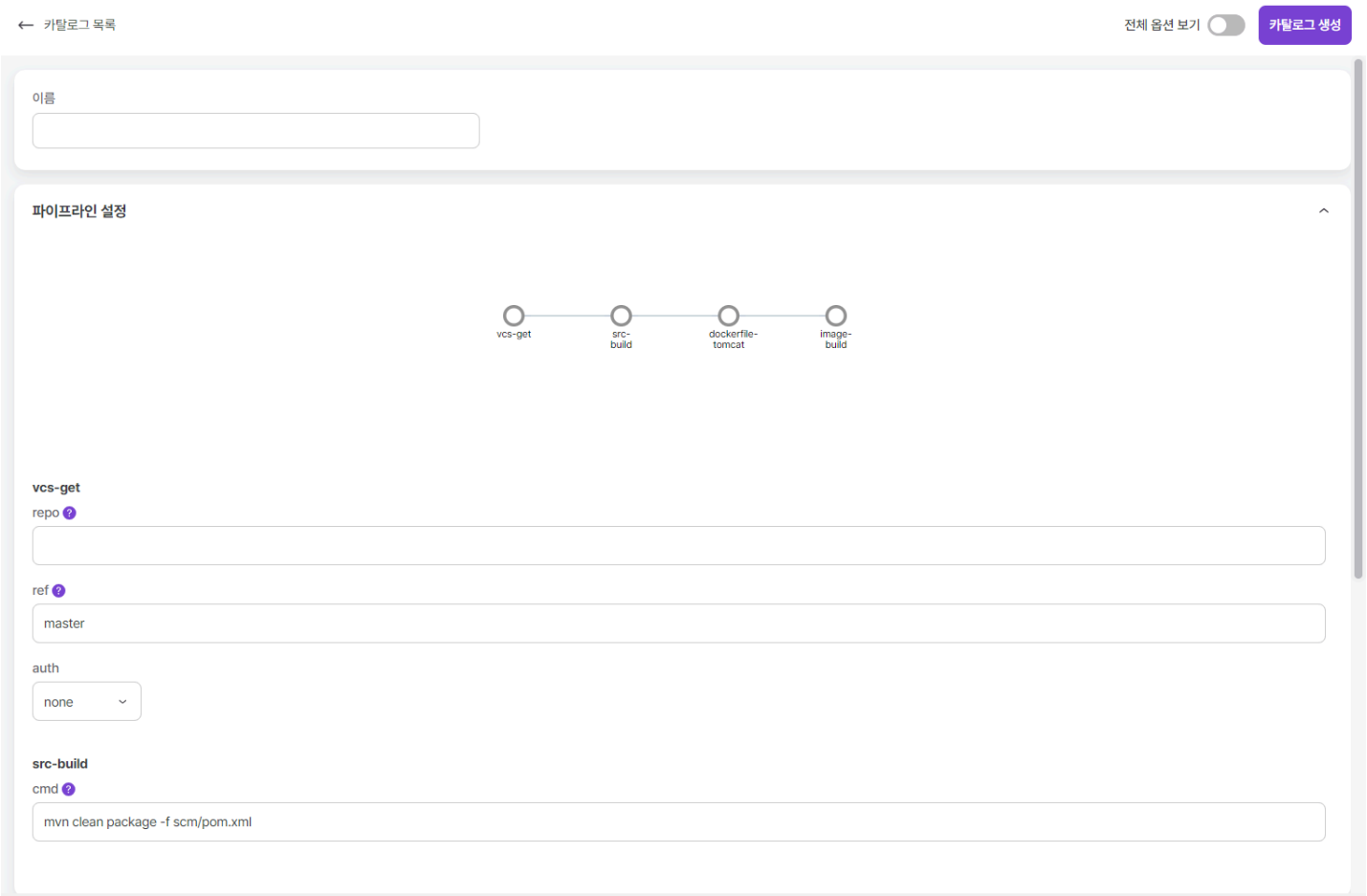



**NOTE**

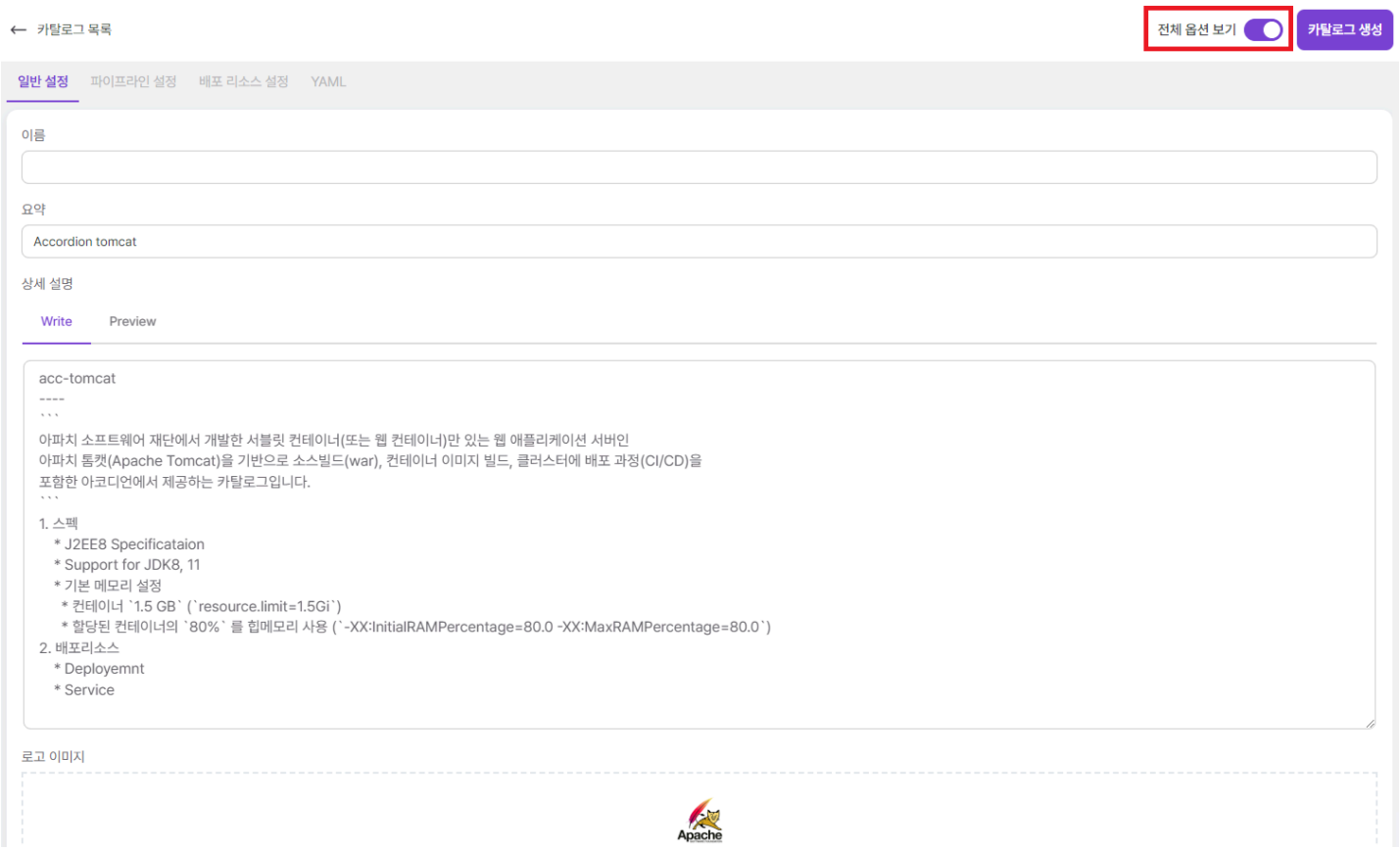
아코디언은 Tomcat, Wildfly 등에 대한 템플릿을 기본 제공하며 카탈로그 템플릿은 추가하거나 변경할 수 있다.

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.




| 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) |

**TIP**

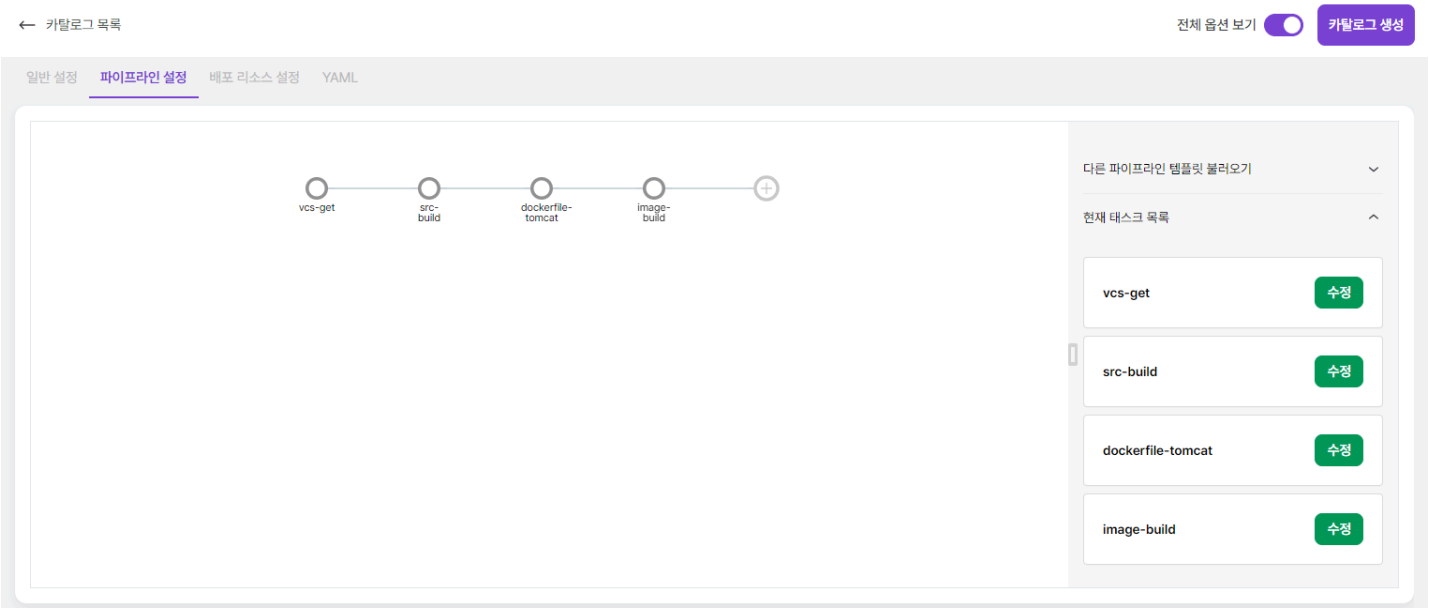
고급 설정 을 선택하면 기본 정보 외 배포 정책 등 상세한 설정이 가능하다. 고급 설정의 항목은 카탈로그마다 조금씩 다를 수 있으며 자주 사용하는 항목은 다음과 같다.

- 배포전략 / 배포 정책 종류

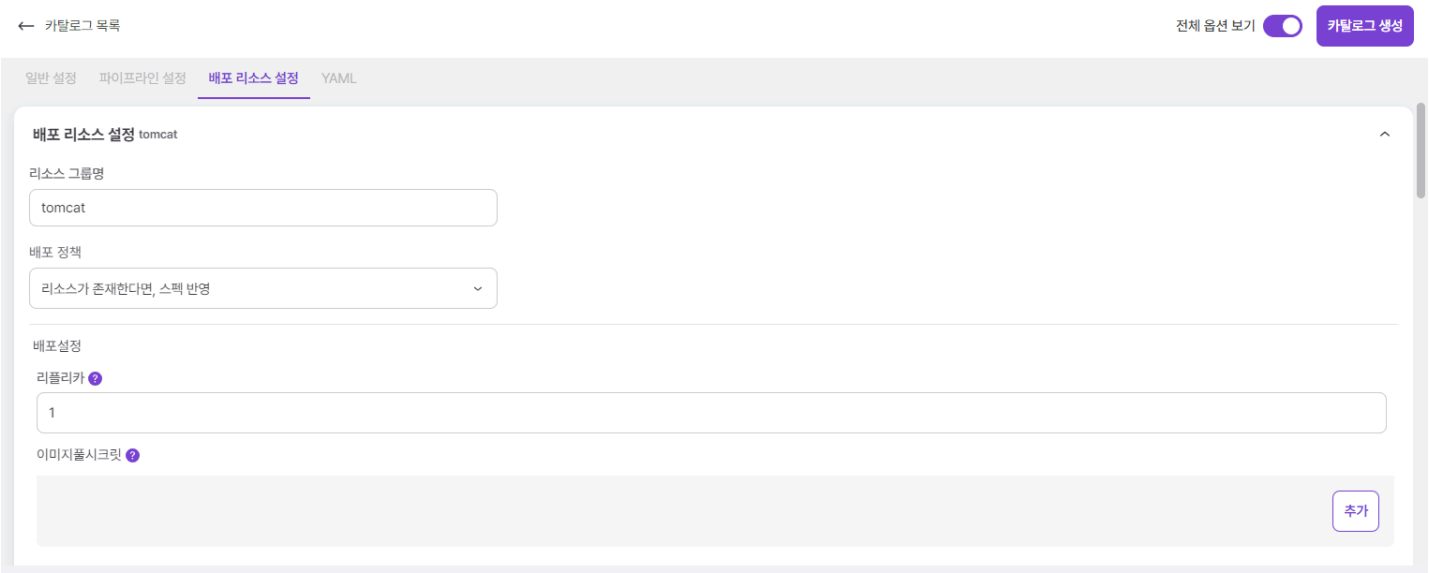
| 정책 | 유사한 명령어 |
| --- | --- |
| 리소스가 존재하면 스펙 반영 | kubectl apply ... |
| 리소스가 존재하면 스펙 대체 | kubectl replace ... |
| 리소스가 존재하면 배포 생략 | kubectl create ... |
| 리소스가 존재하면 대체 우선의 업데이트 | |

- 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.

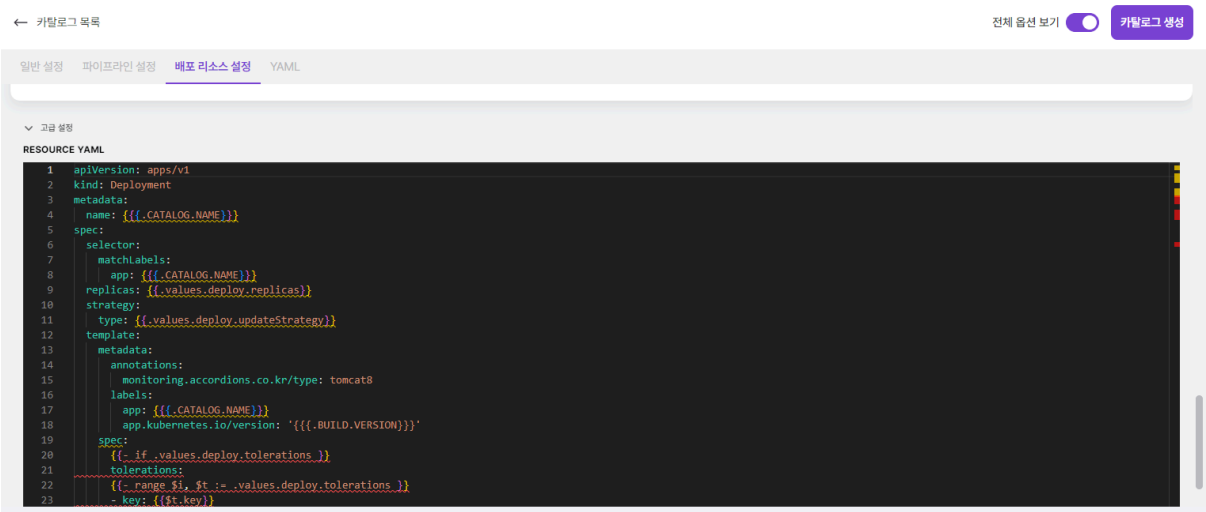



상단 탭 배포 리소스 설정 버튼으로 쿠버네티스에 배포할 리소스에 대한 설정화면으로 변경할 수 있다. 배포 리소스 설정에서는 환경변수 또는 시스템 리소스 할당과 같이 카탈로그로 배포하는 쿠버네티스 리소스에 대한 정보를 입력한다.

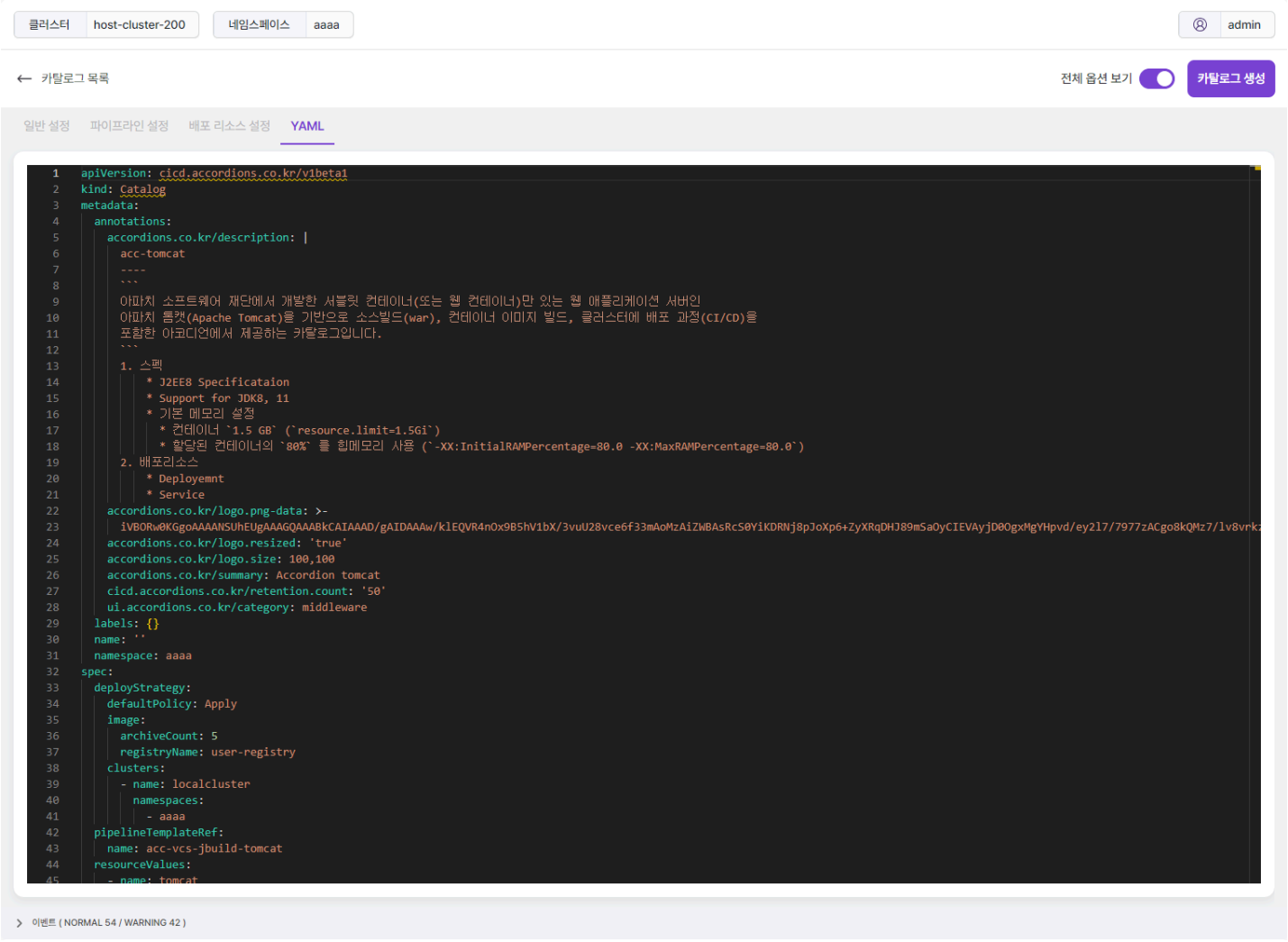


**TIP**

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



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




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}}} | • catalog.spec.deployStrategy.image.registryName<br>• 카탈로그를 사용하지 않는 경우 공백의 기본값 |
| 이미지 이름 | {{{.IMAGE.NAME}}} | • {{{.IMAGE.REGISTRYNAME}}} 가 존재하는 경우 설정<br>• 이름 생성 규칙은 하단 참고<br>• 카탈로그를 사용하지 않는 경우 공백의 기본값 |
| 이미지 태그 | {{{.IMAGE.TAG}}} | • 명시적으로 입력하면, {{{.IMAGE.NAME}}} 에서 태그 고정 |

| 이름 | 환경변수 표현식 | 추가설명 |
| --- | --- | --- |
| 이미지 이름 패턴 | {{{.IMAGE.NAMEPATTERN}}} | • gen-full (기본값)<br>◦ {{{.IMAGE.NAME}}} 전체 자동 설정<br>◦ 태그는 {{{.BUILD.VERSION}}} 으로 설정되며, 다시빌드( recreate )시, 태그를 현재 버전으로 변경<br>◦ 이 경우에만 보관개수 ( catalog.spec.deployStrategy.image.archiveCount )에 따른 삭제 수행<br>• gen-name<br>◦ {{{.IMAGE.TAG}}} 가 존재하여 {{{.IMAGE.NAME}}} 태그를 제외한 이름 자동 설정<br>• custom<br>◦ 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.

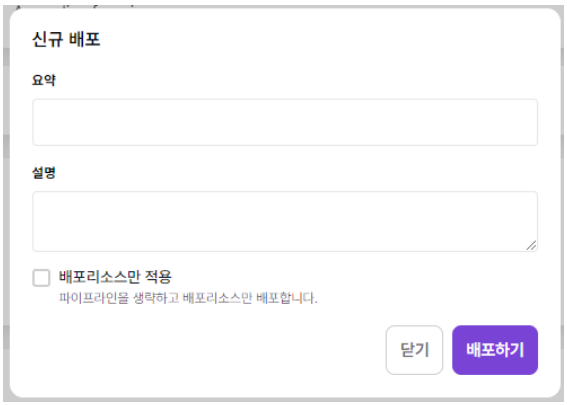
이미지 이름의 생성 규칙과 관련 환경변수 설정은 다음과 같다.

| INPUT | OUTPUT |
| --- | --- |
| 일반 형식<br>• (registry) server : 127.0.0.1:5000<br>• (registry) username : acc<br>• {{{.PIPELINE.INSTANCE}}} : tmapp-c834f245<br>• {{{.BUILD.VERSION}}} : 2 | • {{{.REGISTRY.SERVER}}} : 127.0.0.1:5000<br>• {{{.REGISTRY.USERNAME}}} : acc<br>• {{{.IMAGE.NAMEPATTERN}}} : gen-full<br>• {{{.IMAGE.NAME}}} : 127.0.0.1:5000/acc/tmapp-c834f245:2 |
| Dockerhub<br>• (registry) server : https://index.docker.io/v1/<br>• (registry) username : acc<br>• {{{.PIPELINE.INSTANCE}}} : tmapp-c834f245<br>• {{{.BUILD.VERSION}}} : 2 | • {{{.REGISTRY.SERVER}}} : docker.io<br>• {{{.REGISTRY.USERNAME}}} : acc<br>• {{{.IMAGE.NAMEPATTERN}}} : gen-full<br>• {{{.IMAGE.NAME}}} : docker.io/acc/tmapp-c834f245:2 |

| INPUT | OUTPUT |
| --- | --- |
| 스코프를 포함하는 경우<br>• (registry) server : registry.gitlab.com/scope/test<br>• (registry) username : acc<br>• {{{.PIPELINE.INSTANCE}}} : tmapp-c834f245<br>• {{{.BUILD.VERSION}}} : 2 | • {{{.REGISTRY.SERVER}}} : registry.gitlab.com/scope/test<br>• {{{.REGISTRY.USERNAME}}} : acc<br>• {{{.IMAGE.NAMEPATTERN}}} : gen-full<br>• {{{.IMAGE.NAME}}} : registry.gitlab.com/scope/test/tmapp-c834f245:2 |
| 이미지 태그를 사용하는 경우<br>• (registry) server : 127.0.0.1:5000<br>• (registry) username : acc<br>• {{{.PIPELINE.INSTANCE}}} : tmapp-c834f245<br>• {{{.BUILD.VERSION}}} : 2<br>• {{{.IMAGE.TAG}}} : dev | 이미지 이름의 태그 변경 : 빌드버전 ⇒ 태그<br>• {{{.REGISTRY.SERVER}}} : 127.0.0.1:5000<br>• {{{.REGISTRY.USERNAME}}} : acc<br>• {{{.IMAGE.TAG}}} : dev<br>• {{{.IMAGE.NAMEPATTERN}}} : gen-name<br>• {{{.IMAGE.NAME}}} : 127.0.0.1:5000/acc/tmapp-c834f245:dev |
| 이미지 이름을 사용하는 경우<br>• (registry) server : 127.0.0.1:5000<br>• (registry) username : acc<br>• {{{.PIPELINE.INSTANCE}}} : tmapp-c834f245<br>• {{{.IMAGE.NAME}}} : 127.0.0.1:5000/change/custom-app:latest | 설정한 이미지 이름 고정<br>• {{{.REGISTRY.SERVER}}} : 127.0.0.1:5000<br>• {{{.REGISTRY.USERNAME}}} : acc<br>• {{{.IMAGE.NAMEPATTERN}}} : custom<br>• {{{.IMAGE.NAME}}} : 127.0.0.1:5000/change/custom-app:latest |
| 이미지 이름과 이미지 태그를 같이 사용하는 경우<br>• (registry) server : 127.0.0.1:5000<br>• (registry) username : acc<br>• {{{.PIPELINE.INSTANCE}}} : tmapp-c834f245<br>• {{{.IMAGE.TAG}}} : dev<br>• {{{.IMAGE.NAME}}} : 127.0.0.1:5000/change/custom-app:latest | 이미지 태그 무시<br>• {{{.REGISTRY.SERVER}}} : 127.0.0.1:5000<br>• {{{.REGISTRY.USERNAME}}} : acc<br>• {{{.IMAGE.NAMEPATTERN}}} : custom<br>• {{{.IMAGE.NAME}}} : 127.0.0.1:5000/change/custom-app:latest |

모든 설정이 끝나면 `생성` 버튼을 선택하여 카탈로그 생성을 마친다. 카탈로그가 생성되었다고하여 실제 애플리케이션이 배포된 것은 아니다. 카탈로그 목록에서 방금 작성한 카탈로그를 선택하고 `배포하기` 버튼을 선택해야 파이프라인을 통해 애플리케이션이 배포된다.

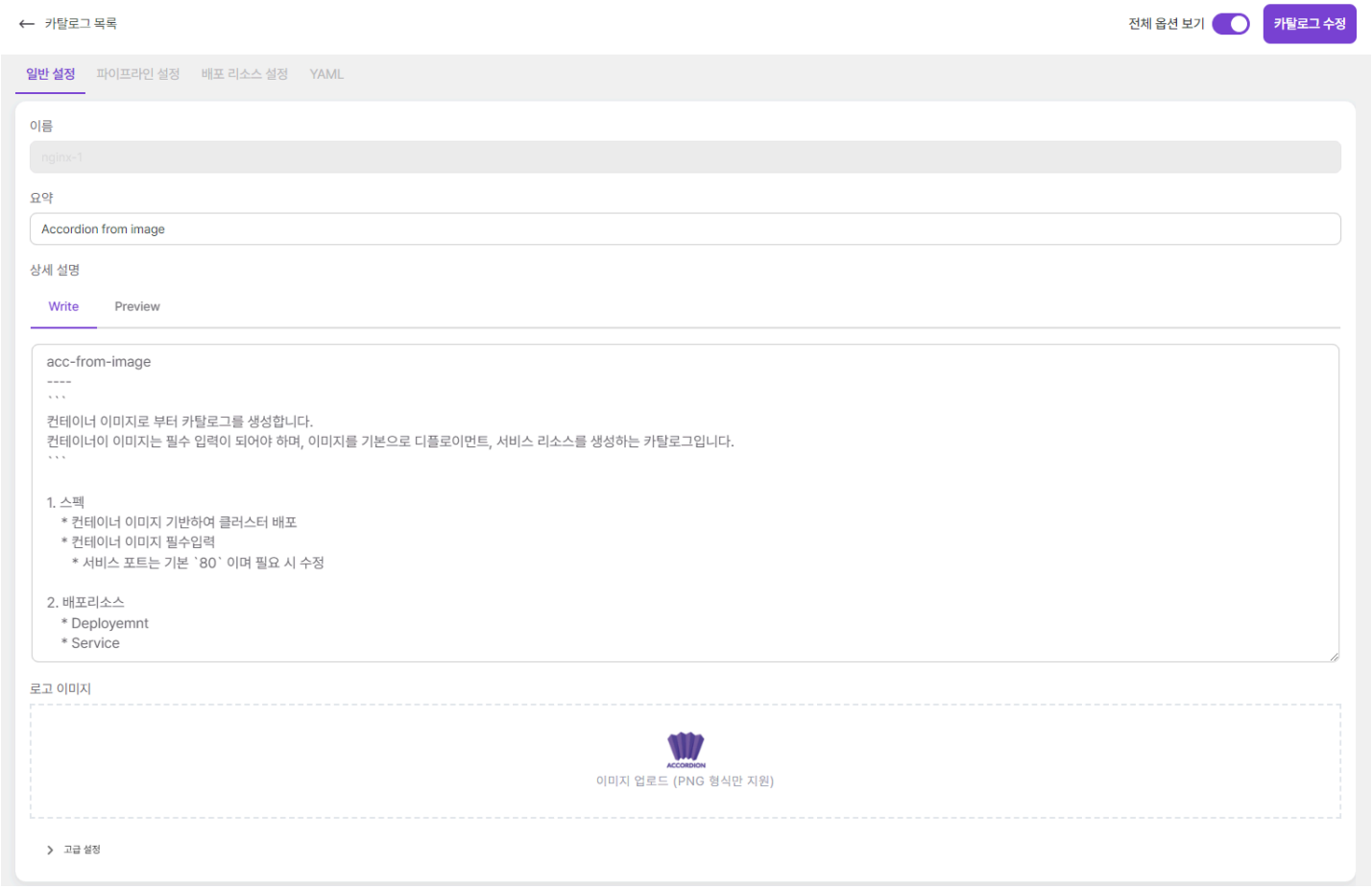
`배포리소스만 적용` 을 선택하여 배포할 경우 가장 최근 배포한 빌드의 이미지를 사용하여 현재 카탈로그에 정의된 스펙으로 리소스 배포만 진행한다. 가장 최근 배포한 빌드의 이미지는 2.9.0 업그레이드 이후 배포한 이미지만 해당한다.
2.9.0 이전 버전의 빌드 이미지로는 `배포리소스만 적용` 하여 배포가 불가능하며, 해당 기능을 사용하려면 업그레이드 이후 최소 1번의 배포를 진행해야 한다.




**TIP** 배포시 요약 과 설명 은 필수값은 아니지만 애플리케이션 관리를 위해 설정을 권장한다.

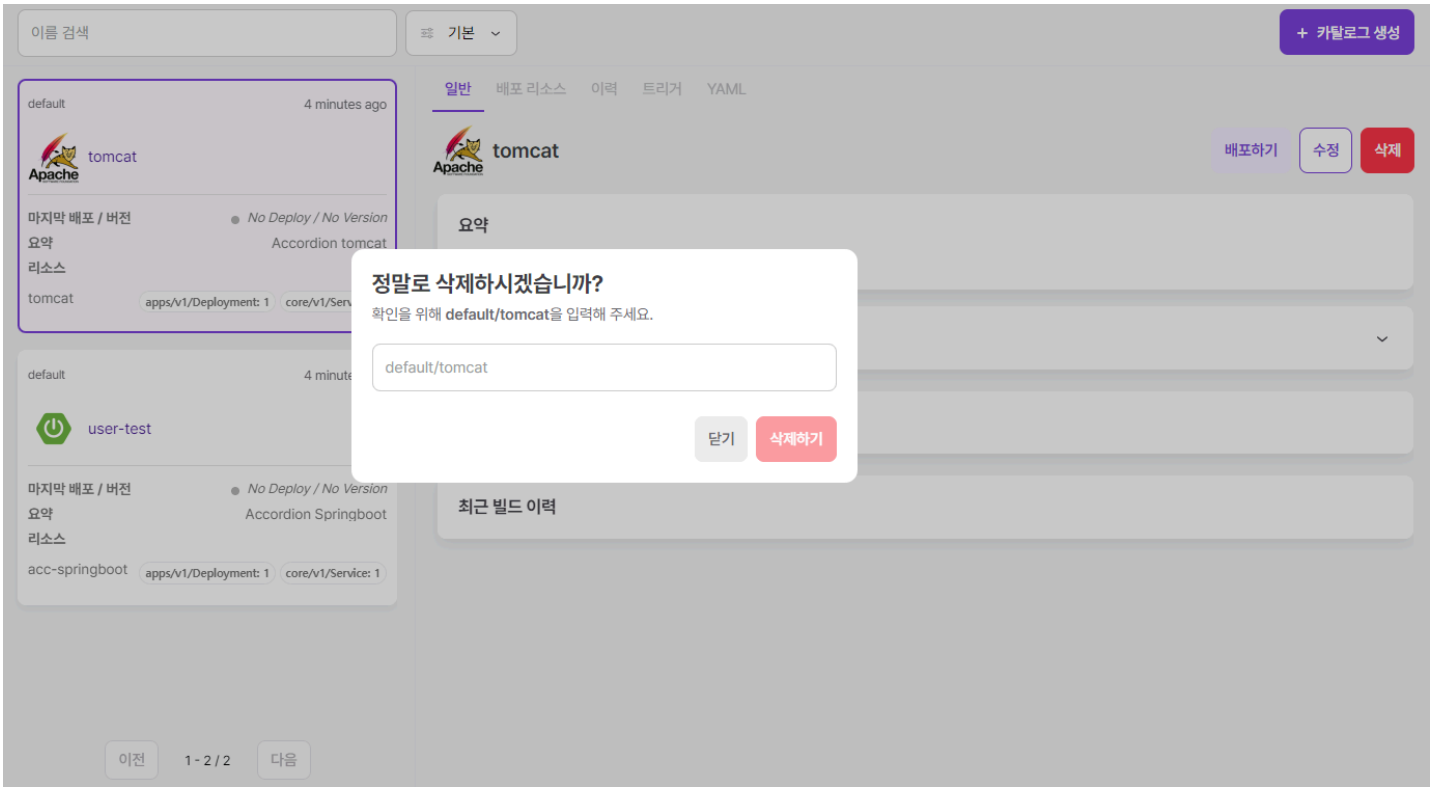
## 카탈로그 수정

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




## 카탈로그 삭제

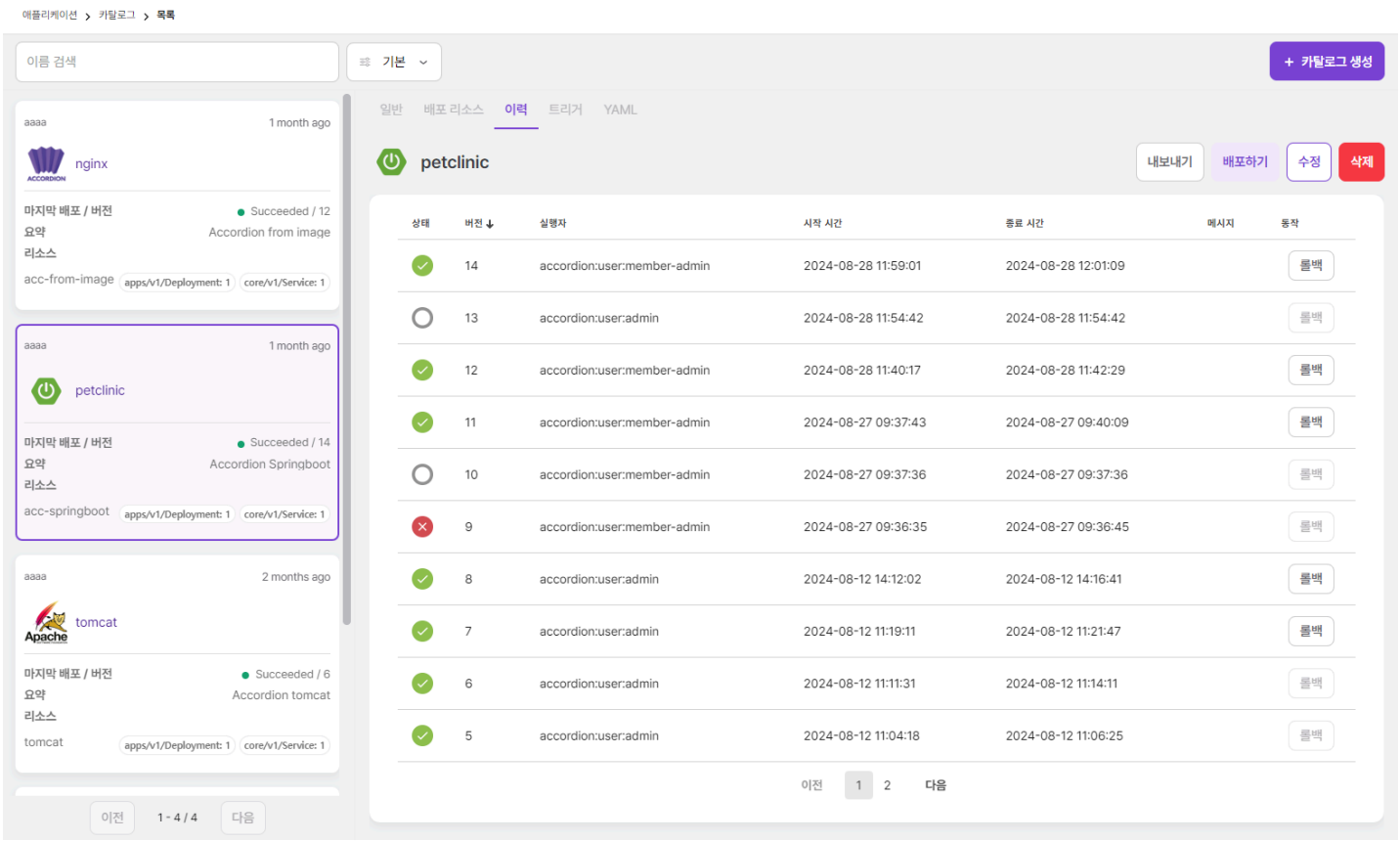
삭제하려는 카탈로그를 선택하고 우측의 `삭제` 버튼을 선택한다.



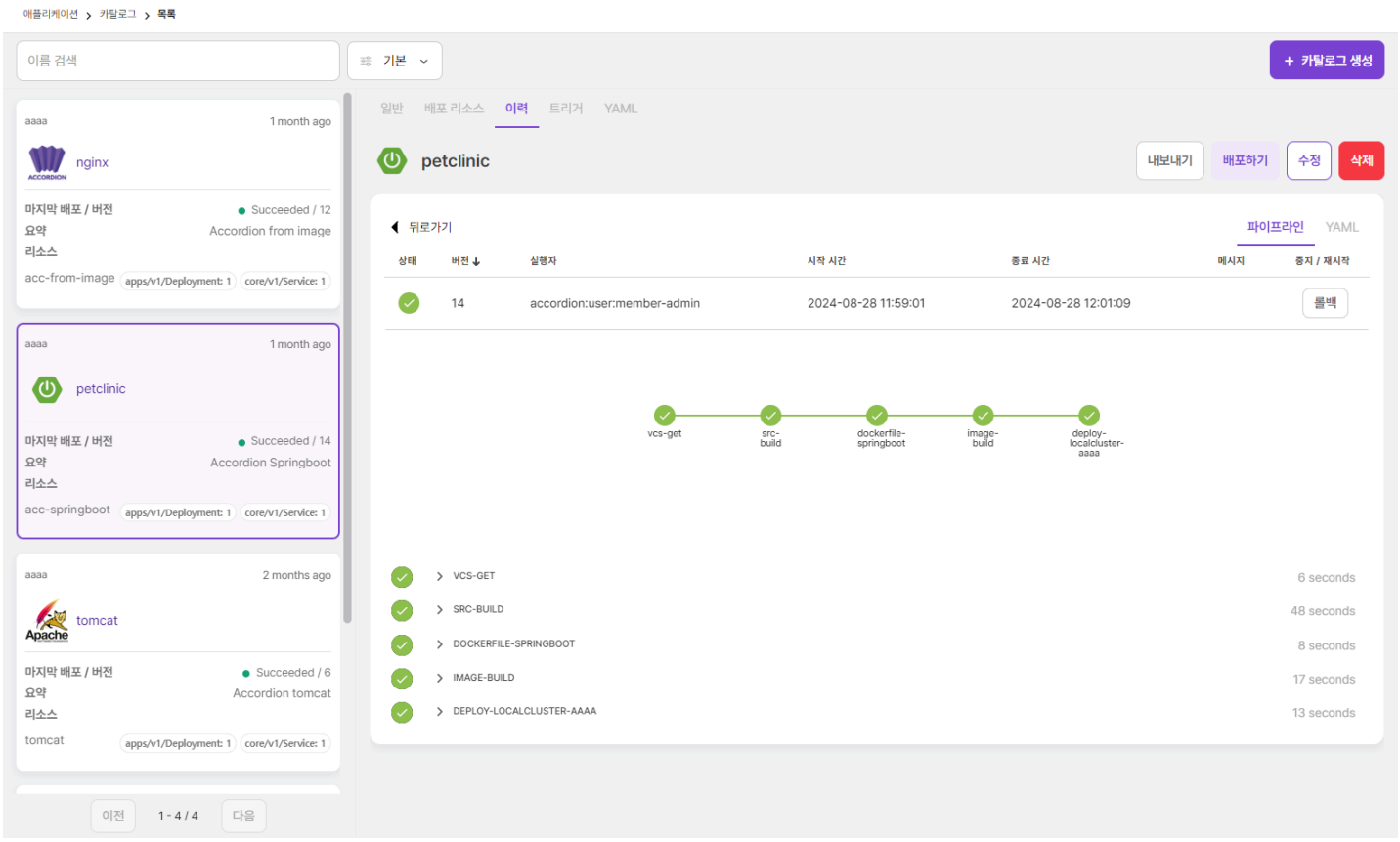
모달에서 네임스페이스와 카탈로그 이름을 입력하여 삭제한다.

## 이력

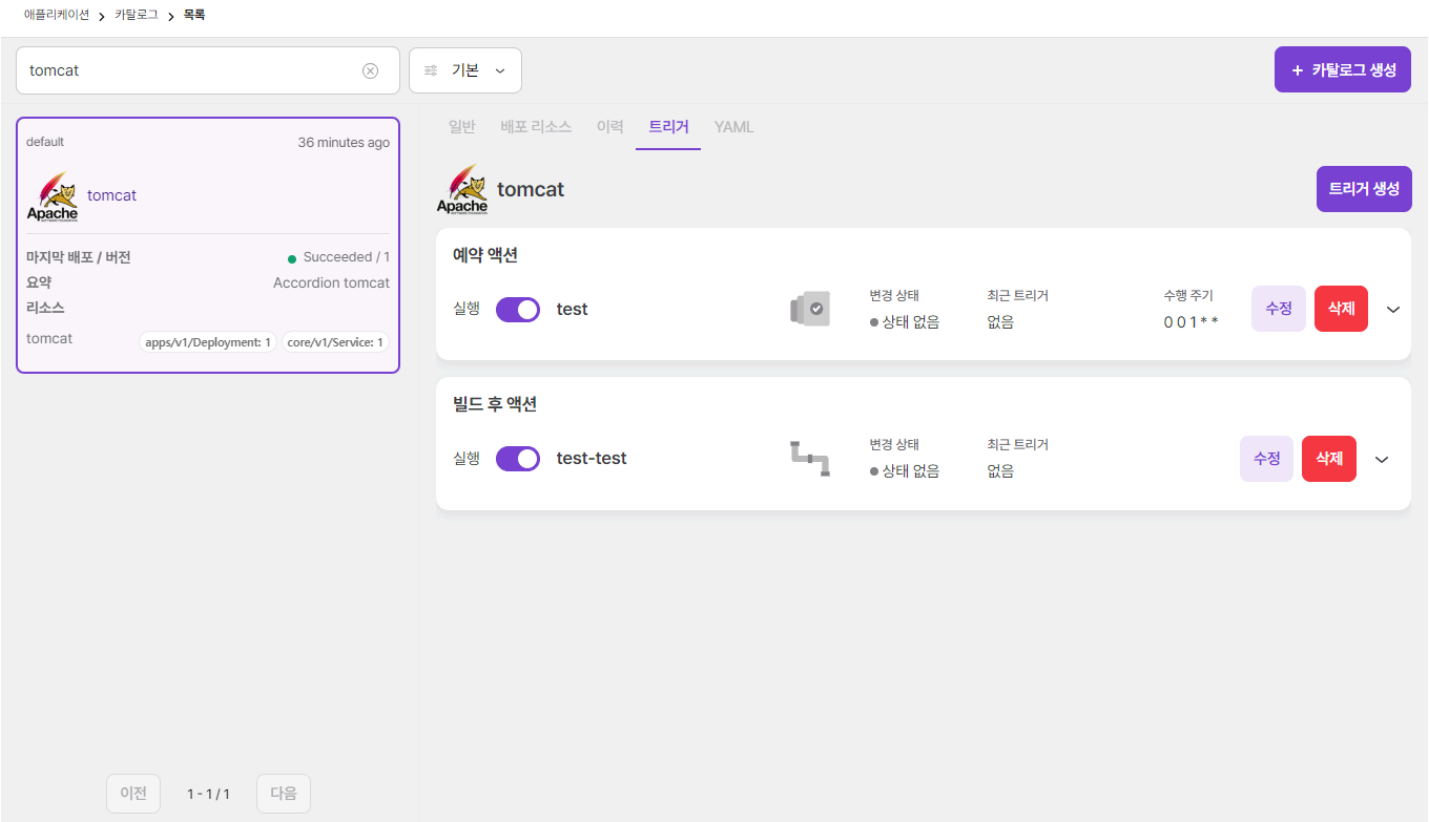
배포한 카탈로그의 이력들을 확인할 수 있다.
특정 이력에 대한 롤백 기능 및 해당 이력들을 csv 파일로 추출하는 내보내기 기능을 제공한다.
롤백의 경우, 배포리소스만 적용 기능을 사용할 수 없다.




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



## 트리거




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 `spec.env.EXTERNAL_IMAGE` 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) 방식으로 등록한다.

- 스케쥴의 경우, **3-5분 이상** 으로 등록하길 권장한다.
- 기본값 : `0/5 * * * *`

버전관리 시스템의 저장소 변경 감지 트리거
vcs 태스크의 메타데이터를 원격 저장소와 비교하여, 변경사항을 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 변경사항을 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 mode가 **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.EXTERNALIMAGE` 정보를 통해 이미지 변경을 감지한다.

YAML
```
spec:
  env:
    EXTERNALIMAGE:
      DIGEST:       sha256:9822b71cd6b28545703fde132adb98cd1057ea387e41c1d84a254486feb3c2cc
      NAME:         image
      REGISTRYNAME: user-registry
```

catalog 의 spec.deployStrategy.externallmage 에 아래와 같이 등록할 정보를 입력하고 빌드 또는 배포하면 위와 같이 빌드의 spec.env.EXTERNALIMAGE 에 정보가 등록된다.

YAML
```
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 컨텐츠는 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.

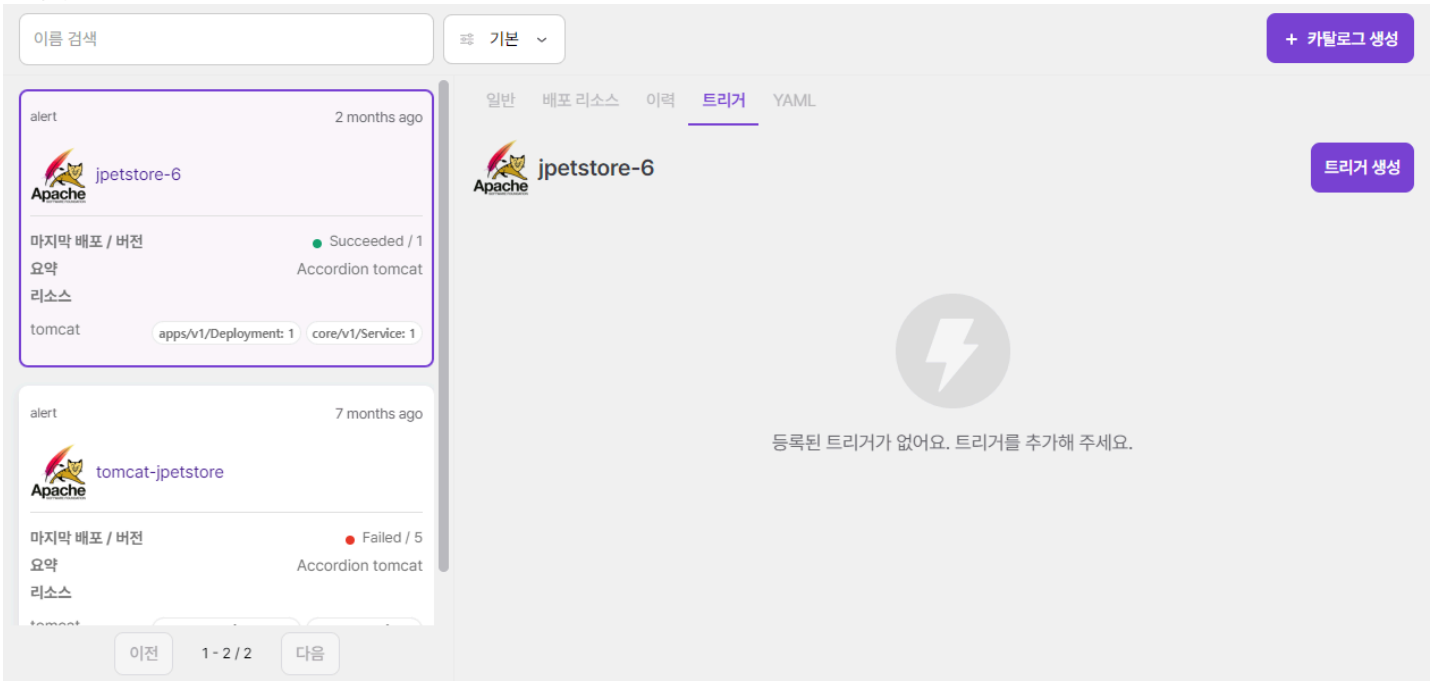
| 트리거 이름 | 설명 |
| --- | --- |
| 카탈로그/파이프라인 빌드(배포) 트리거 | 같은 클러스터의 다른 카탈로그/파이프라인을 빌드(배포) |
| 웹훅 전달 트리거 | 대상 객체와 빌드 정보를 http로 보내준다. |

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

웹훅 전달 트리거
보내주는 데이터 형식은 아래와 같다.

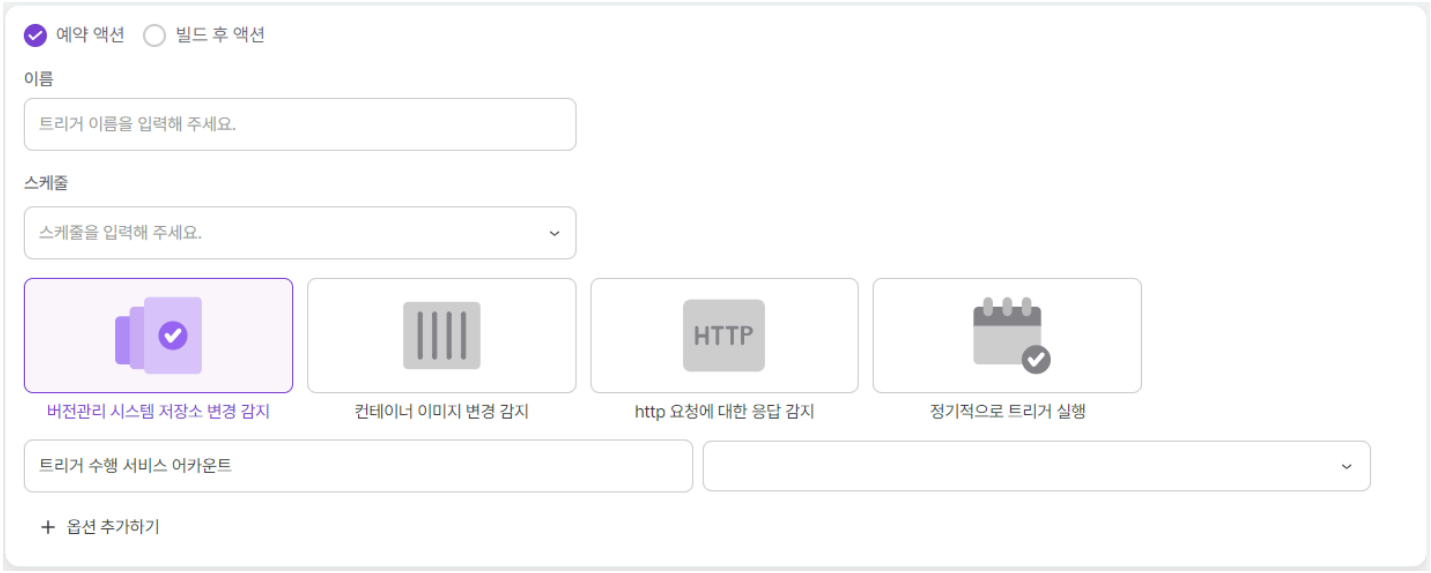
JSON
```
{
  "object": {카탈로그 or 파이프라인},
  "build": {마지막 빌드}
}
```

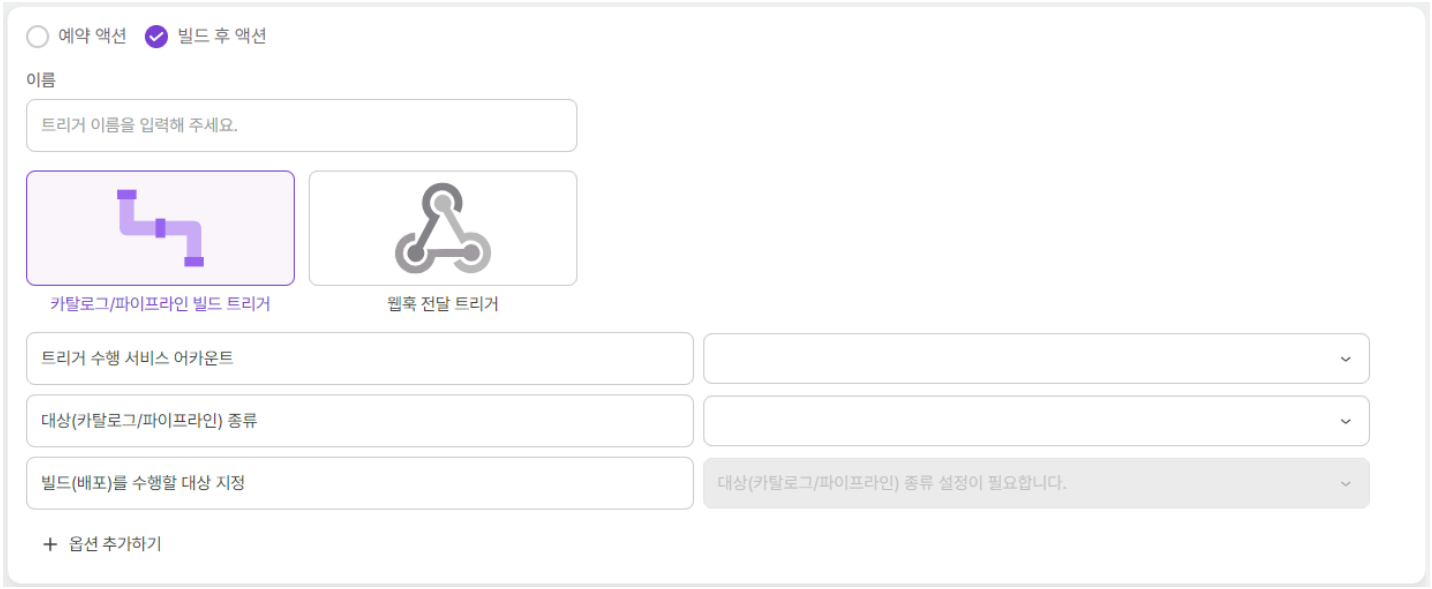
트리거 등록




트리거 탭을 선택한 뒤 트리거 생성 버튼을 클릭

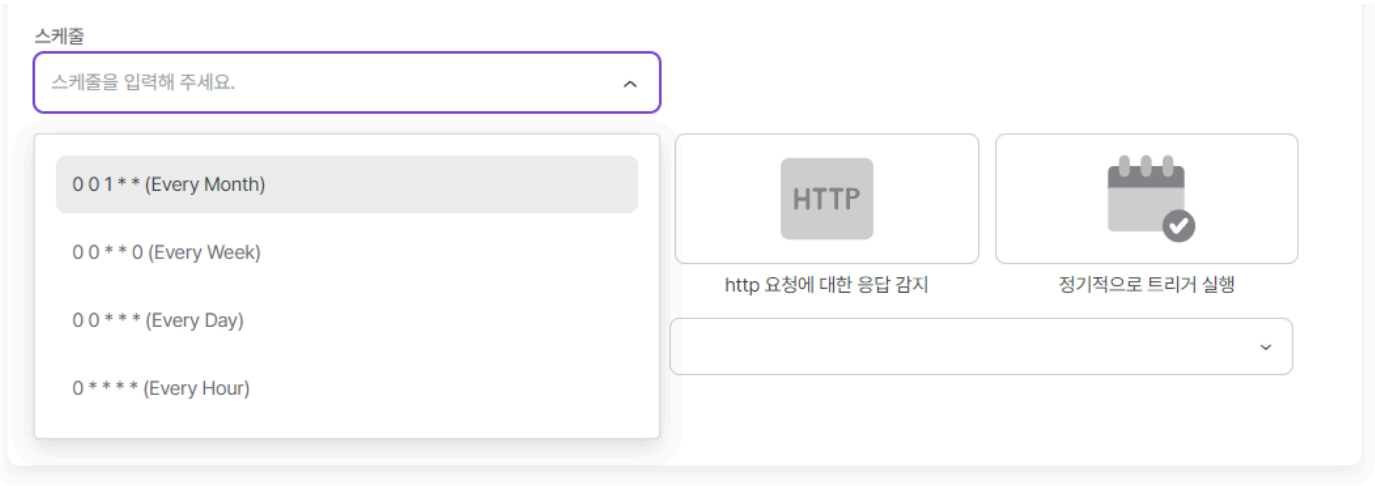
예약 액션 트리거 또는 빌드 후 액션 트리거 를 선택한다.




예약 액션
빌드 후 액션
이름
트리거 이름을 입력해 주세요.
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` 셀렉트박스를 클릭하면 기본으로 설정되어 있는 스케줄을 설정할 수 있다.

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

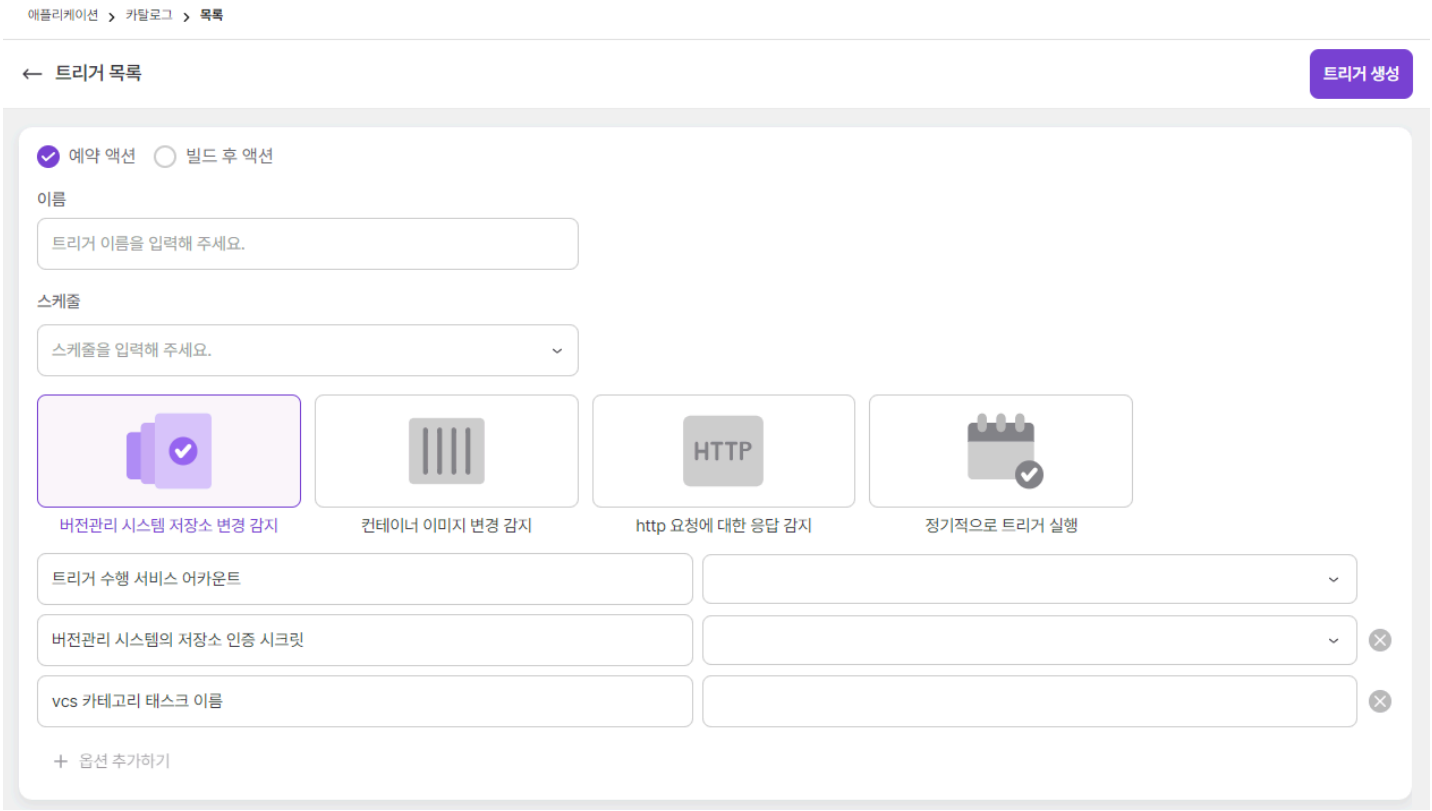
| 시간 | 허용 가능 값 | 허용 가능 특수문자 |
|---|---|---|
| 분 | 0-59 | * / , - |
| 시 | 0-23 | * / , - |
| 일자 | 1-13 | * / , - ? |
| 달 | 1-12 or JAN-DEC | * / , - |
| 요일 | 0-6 or SUN-SAT | * / , - ? |

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.

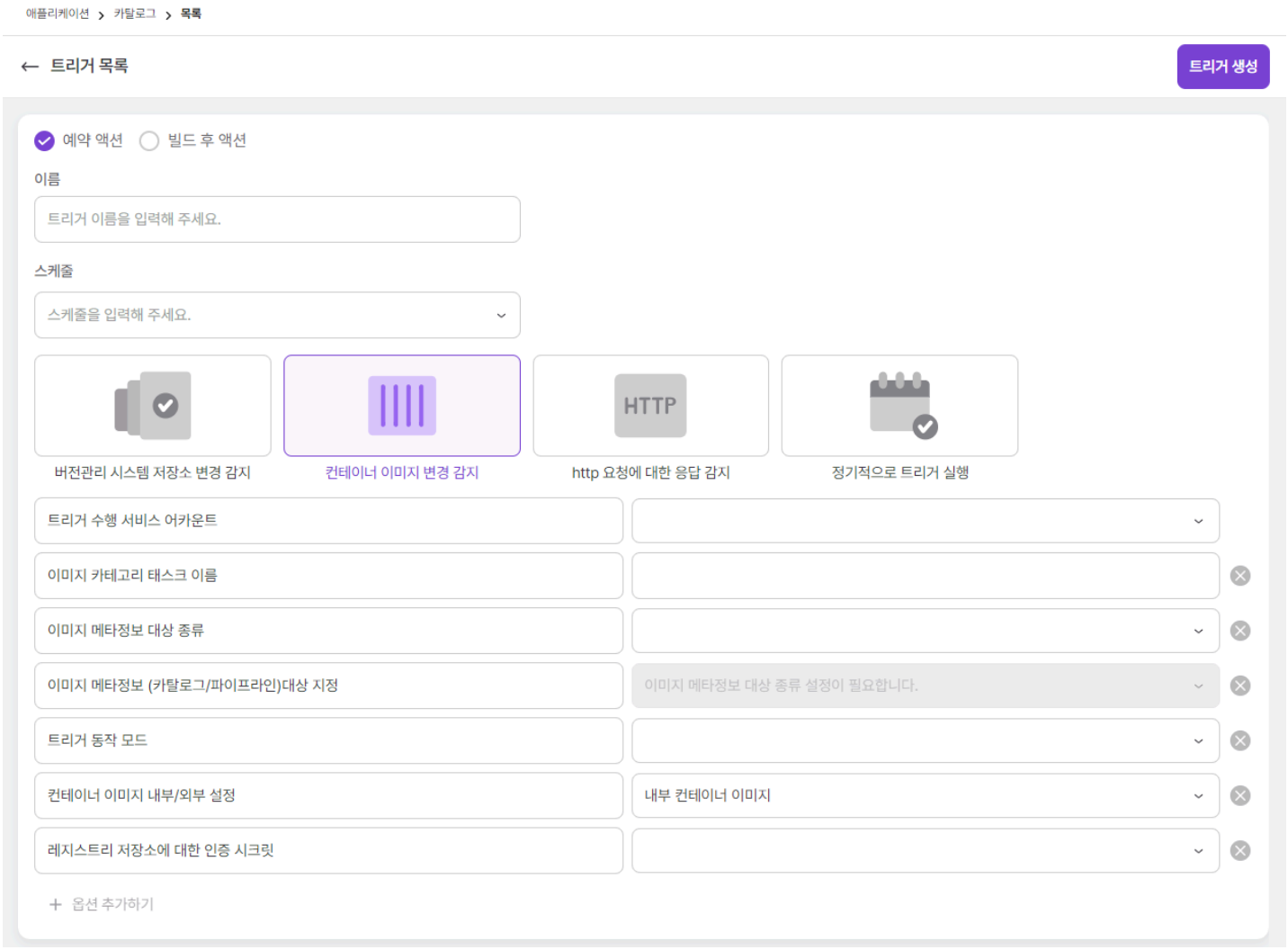
버전관리 시스템의 저장소 변경 감지 트리거 추가




사용 되는 키는 다음과 같다

| 키 | 설명 |
| --- | --- |
| 트리거 수행 서비스 어카운트 | 필수 밸류<br>빌드(배포)를 수행할 인증 서비스 어카운트<br><br>권한이 없다면 빌드(배포)가 실패한다. |
| 버전관리 시스템의 저장소 인증 시크릿 | vcs 저장소에 대한 인증 시크릿<br><br>Task 에 해당하는 태스크 사양(spec)에서 values.auth.cred 값이 있다면 자동 할당<br><br>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) |

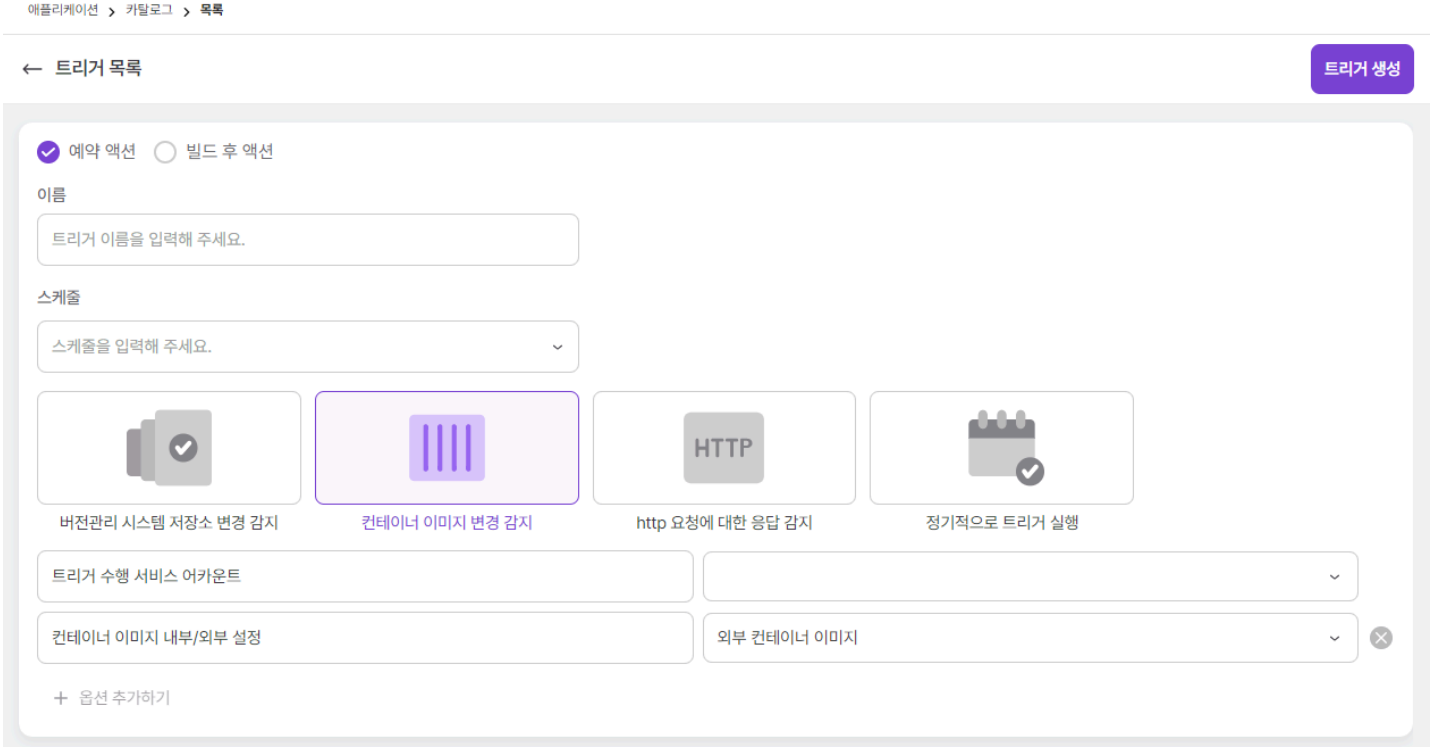
컨테이너 이미지 변경 감지 트리거
내부 이미지




| 키 | 설명 |
| --- | --- |
| 트리거 수행 서비스 어카운트 | 필수 밸류<br><br>빌드(배포)를 수행할 인증 서비스 어카운트<br><br>권한이 없다면 빌드(배포)가 실패한다. |
| 이미지 카테고리 태스크 이름 | 이미지 카테고리 태스크 이름<br><br>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 |

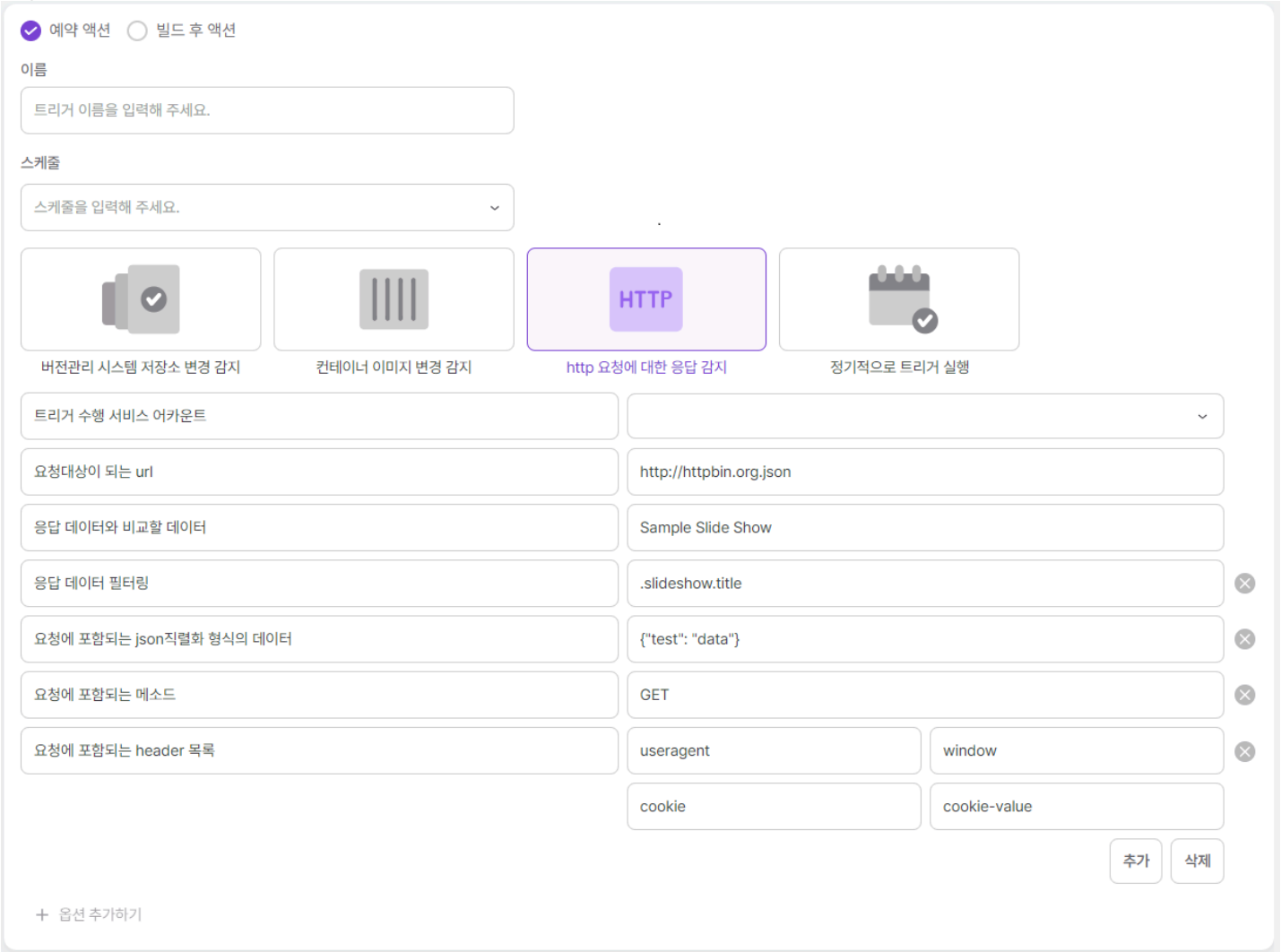
| 키 | 설명 |
| --- | --- |
| 트리거 동작 모드 | digest : 메타데이터의 다이제스트와 레지스트리를 비교하여 변경 여부 확인<br><br>sequence : 태그 버전 비교 방식으로 레지스트리의 마지막 버전 태그가 메타데이터의 버전 태그보다 큰 지 확인<br><br>기본값 : 메타데이터의 이미지 태그가 버전방식이면 sequence 아니면 digest 로 자동 할당<br><br>기본값은 트리거를 등록하는 대상<br><br>네임스페이스를 선택 하고 해당 네임스페이스의 카탈로그/파이프라인 대상을 선택할 수 있다. |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

외부 이미지



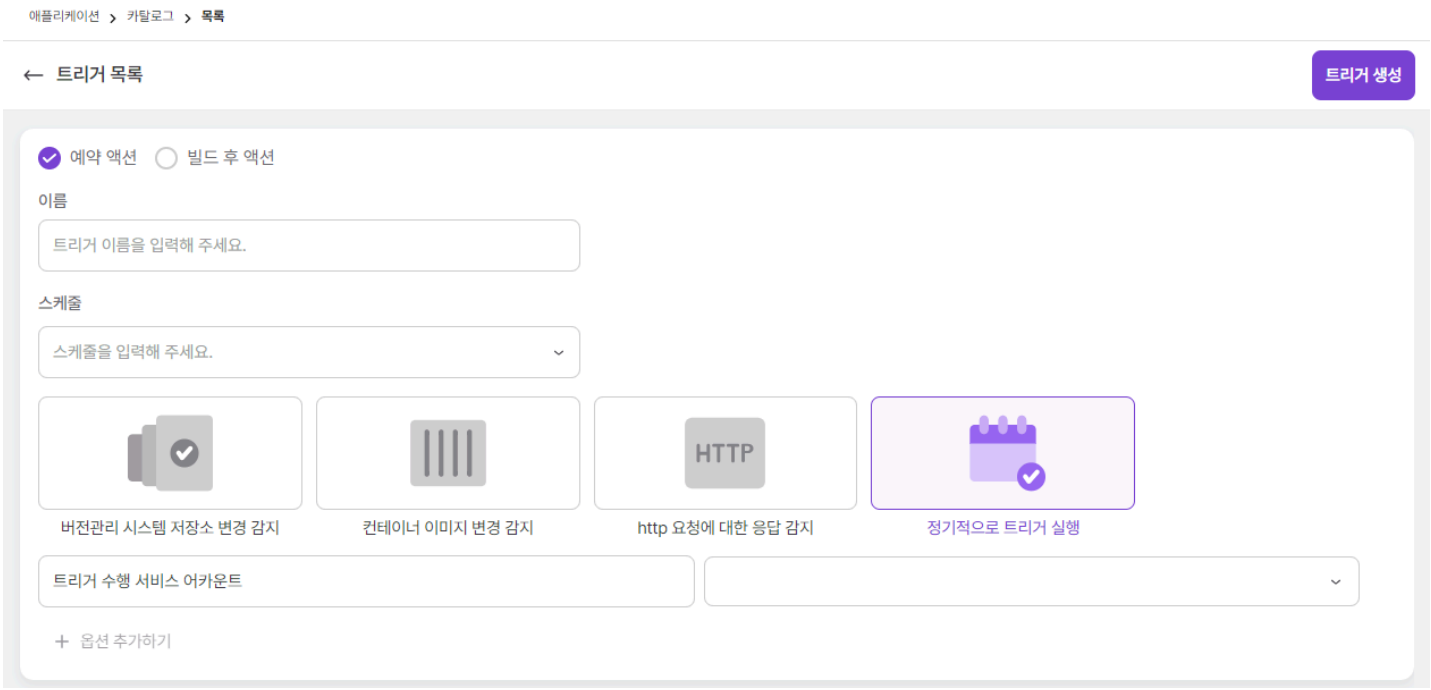

| 키 | 설명 |
| --- | --- |
| 트리거 수행 서비스 어카운트 | 필수 밸류<br><br>빌드(배포)를 수행할 인증 서비스 어카운트<br><br>권한이 없다면 빌드(배포)가 실패한다. |
| 컨테이너 이미지 내부/외부 설정 | 감지할 이미지 대상의 종류를 설정한다. |

http 요청에 대한 응답 감지 트리



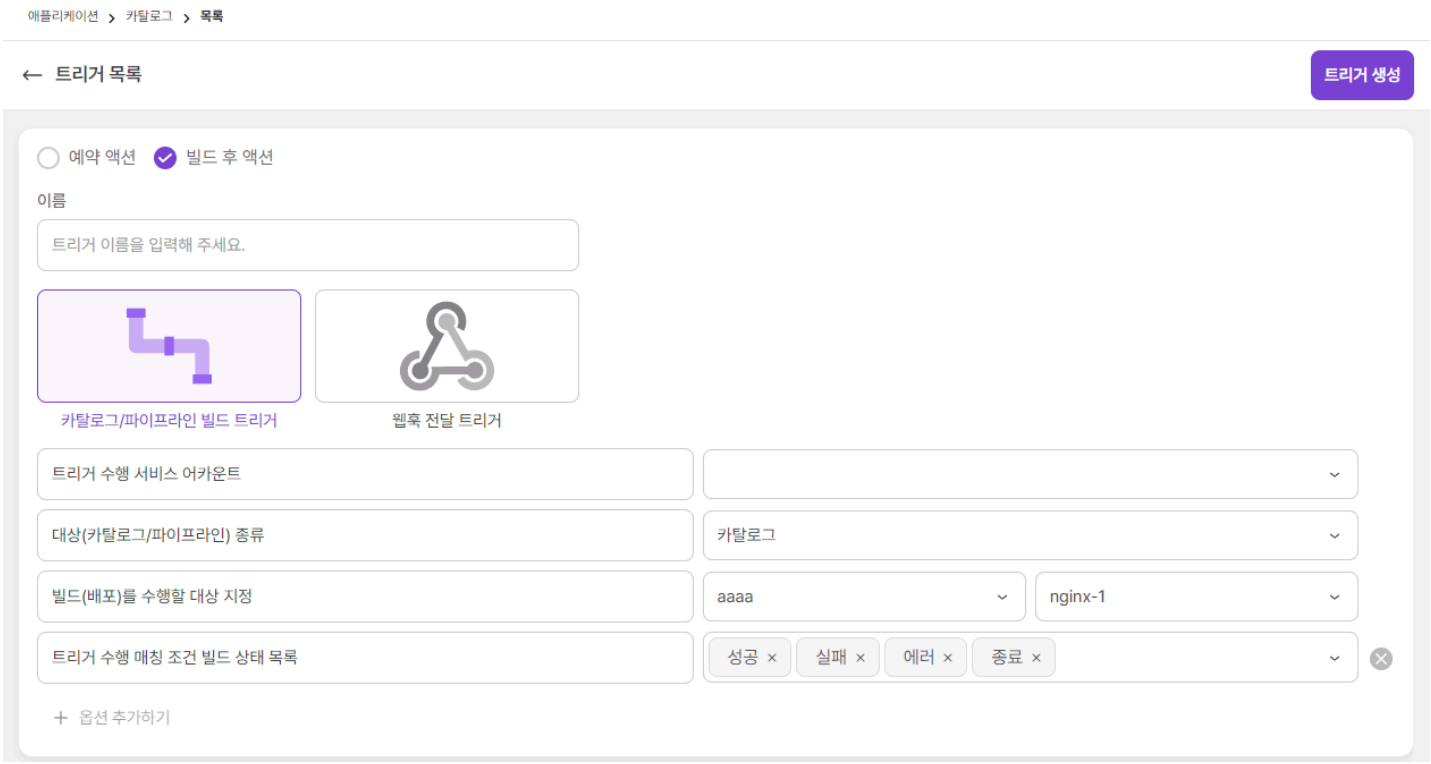

| 키 | 설명 |
| --- | --- |
| 트리거 수행 서비스 어카운트 | 필수 밸류<br><br>빌드(배포)를 수행할 인증 서비스 어카운트<br><br>권한이 없다면 빌드(배포)가 실패한다. |
| 요청대상이 되는 url | 필수 밸류 |
| 응답 데이터와 비교할 데이터 | 필수 밸류<br><br>응답 데이터와 비교할 데이터로 단순 문자열 또는 json직렬화 형식 |
| 응답 데이터 필터링 | 응답 데이터를 필터링하여 응답 데이터와 비교할 데이터 와의 값을 비교<br><br>jsonpath 형식 |
| 요청에 포함되는 json직렬화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

정기적으로 트리거 실행



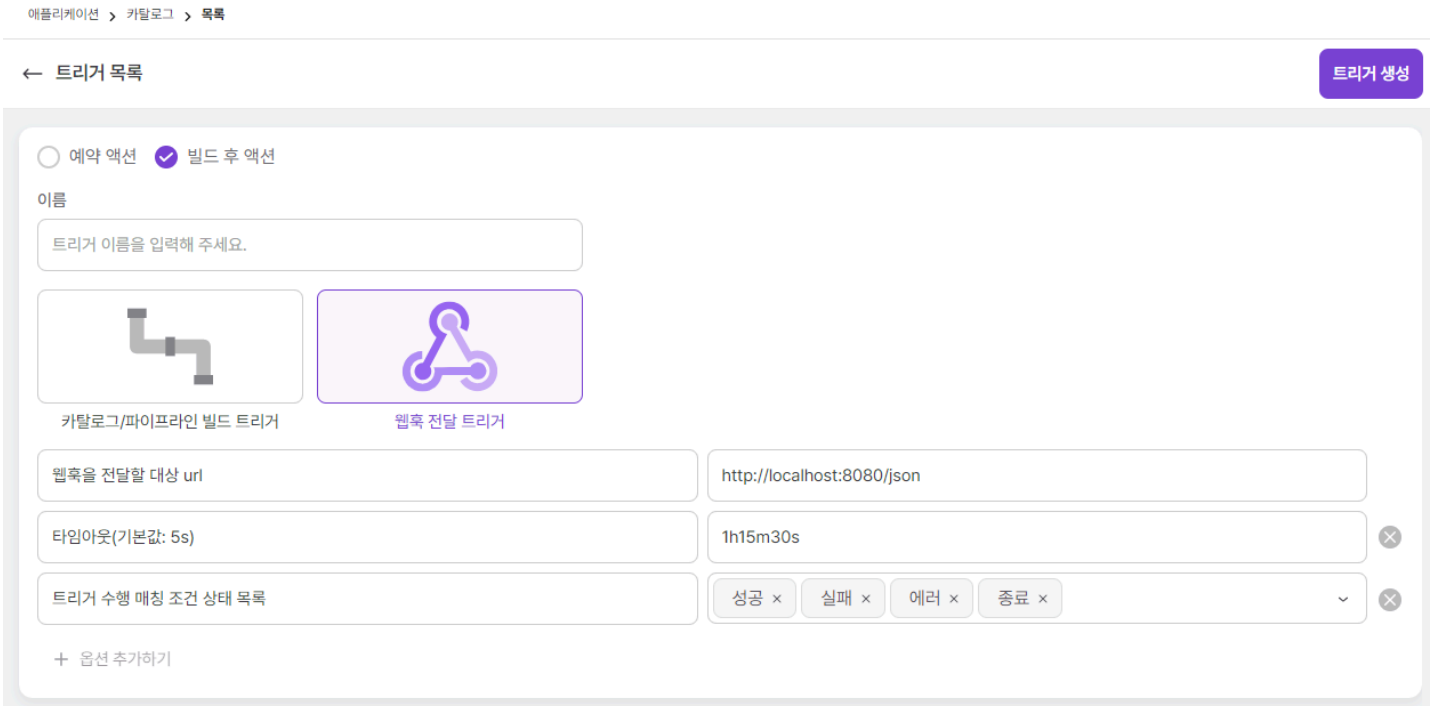

| 키 | 설명 |
| --- | --- |
| 트리거 수행 서비스 어카운트 | 필수 밸류<br><br>빌드(배포)를 수행할 인증 서비스 어카운트<br><br>권한이 없다면 빌드(배포)가 실패한다. |

# 카탈로그/파이프라인 빌드(배포) 트리거




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

웹훅 전달 트리거




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

트리거 수행 서비스 어카운트 생성

**CAUTION** 클러스터 롤 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

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

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.

YAML
```
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
```

- 롤 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 롤과 클러스터 롤 바인딩 생성

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

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

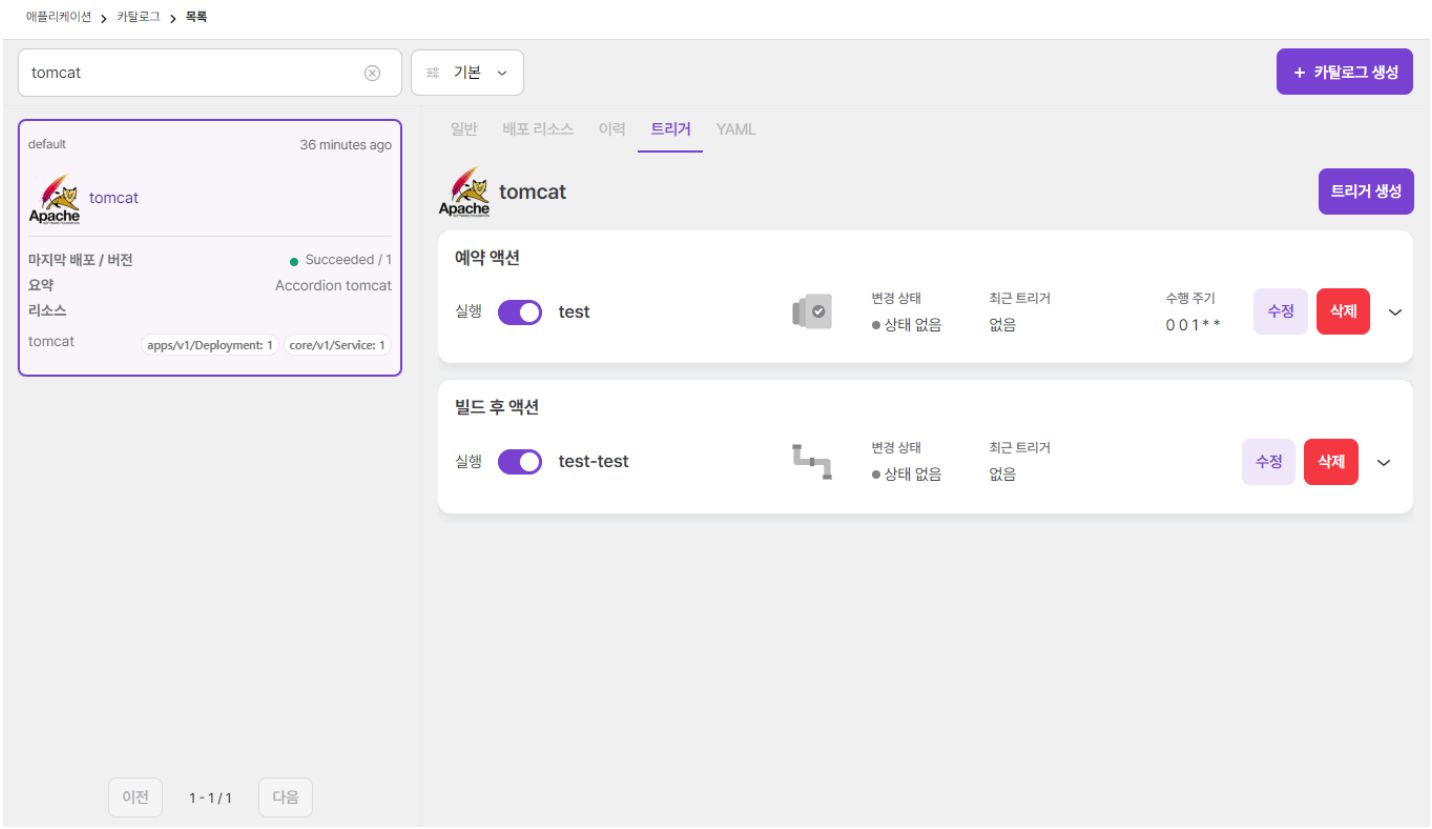
- 클러스터 롤 생성 예시
  - 기본으로 빌드(배포) 하는데 사용되는 권한이며, 빌드(배포) 리소스에 대한 권한이 추가적으로 필요한 경우 기본으로 사용되는 권한과 필요한 권한을 추가하여 사용한다.

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 롤 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




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 정보 이며 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

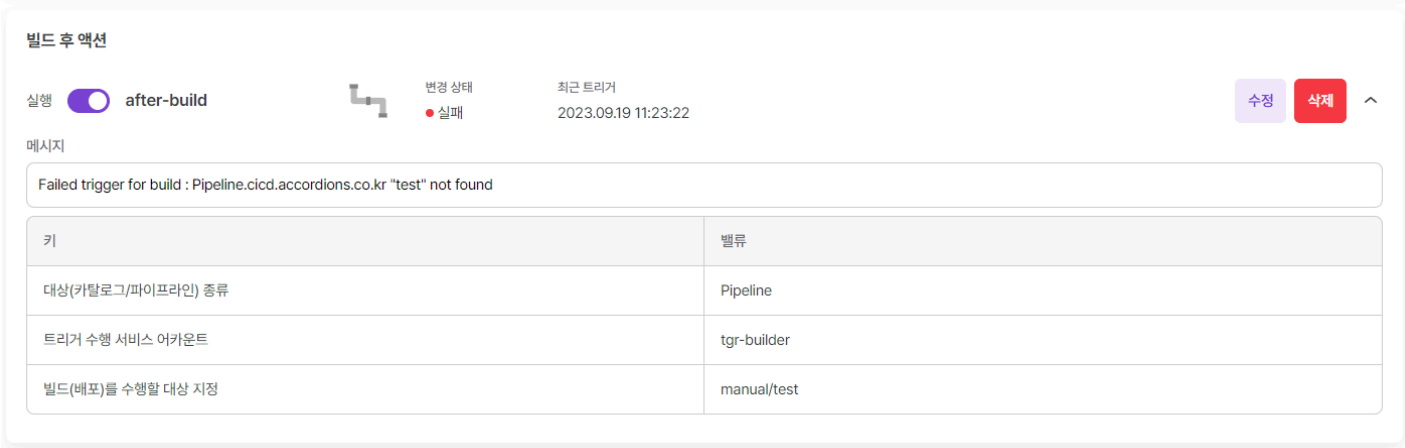
원하는 트리거를 클릭 하면 상세 정보를 확인할 수 있다.

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.

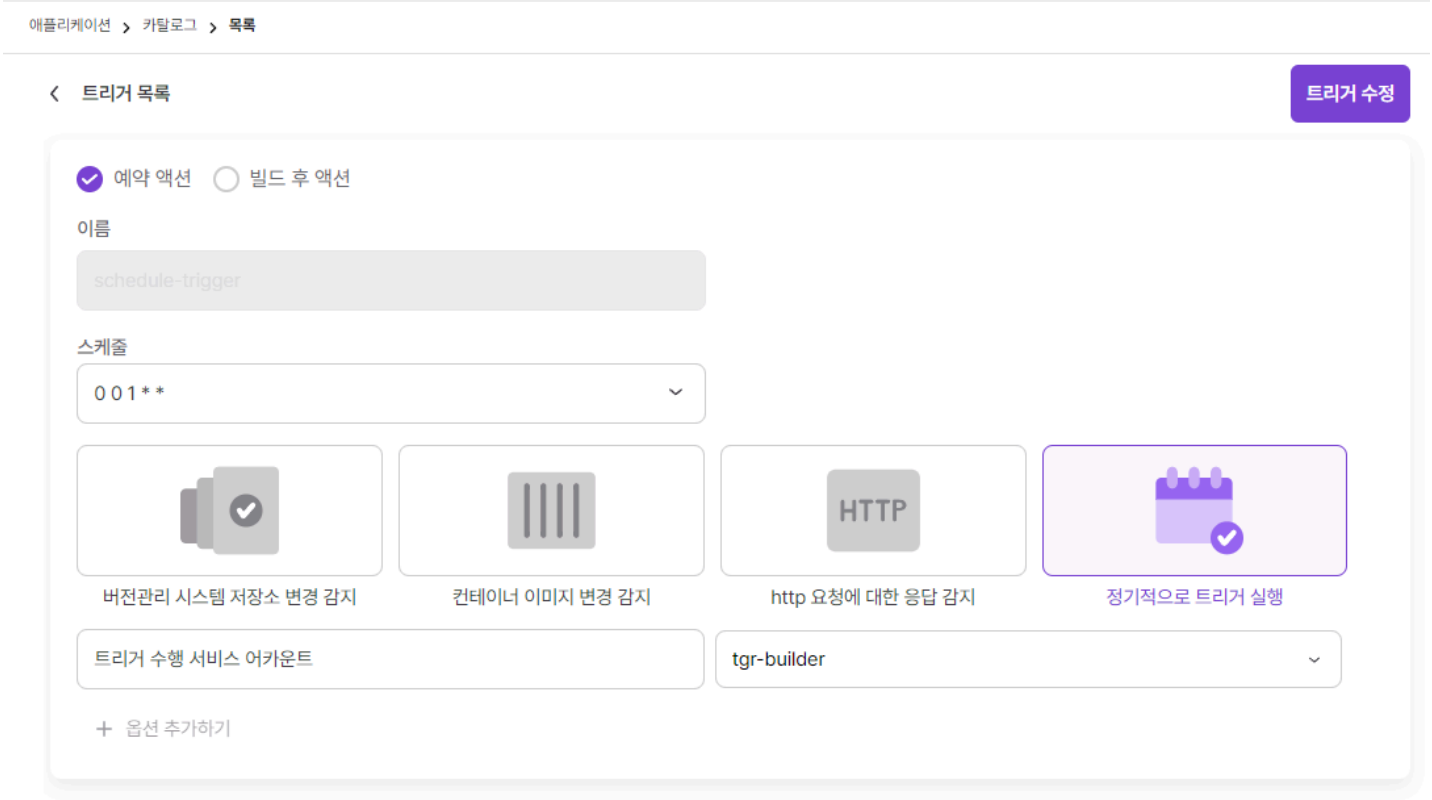
| 종류 | 설명 |
| --- | --- |
| 미감지 | 예약 액션 트리거 에서만 사용되며 트리거가 감지되지 않은 경우 |

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




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

트리거 수정

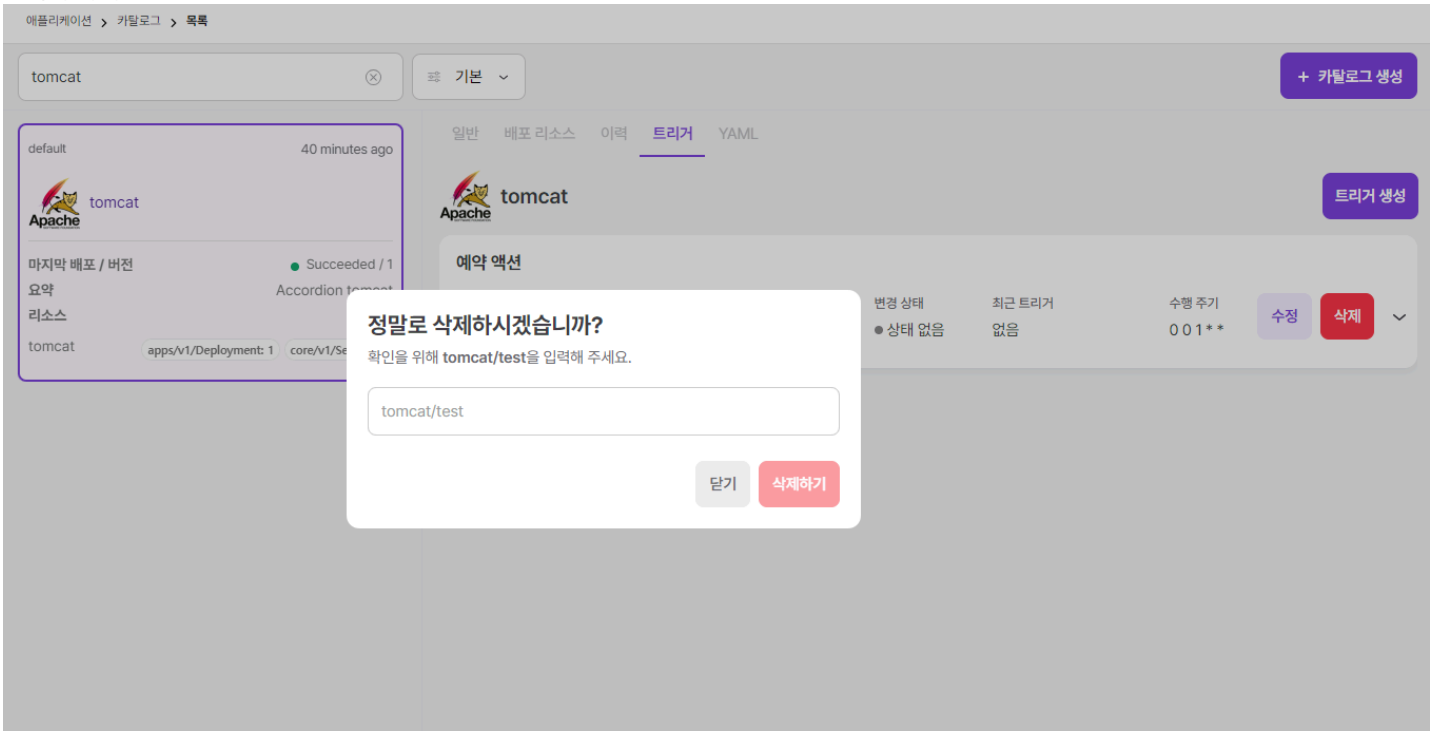



트리거 우측의 수정 버튼을 클릭 하면 수정을 할 수 있는 상태가 된다.

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




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 |
|---|---|
| cicd.accordions.co.kr/retention.count | 개수로 보관할 빌드의 개수를 정한다. |
| cicd.accordions.co.kr/retention.period | 기간으로 보관할 빌드의 기간을 정한다. |

## 등록

YAML

```yaml
metadata:
  annotations:
    cicd.accordions.co.kr/retention.count: '1'
    cicd.accordions.co.kr/retention.period: 1m
```

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

| 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 **1**인 경우 **1 개**의 최신 빌드만 보관한다.

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

## 프로세스

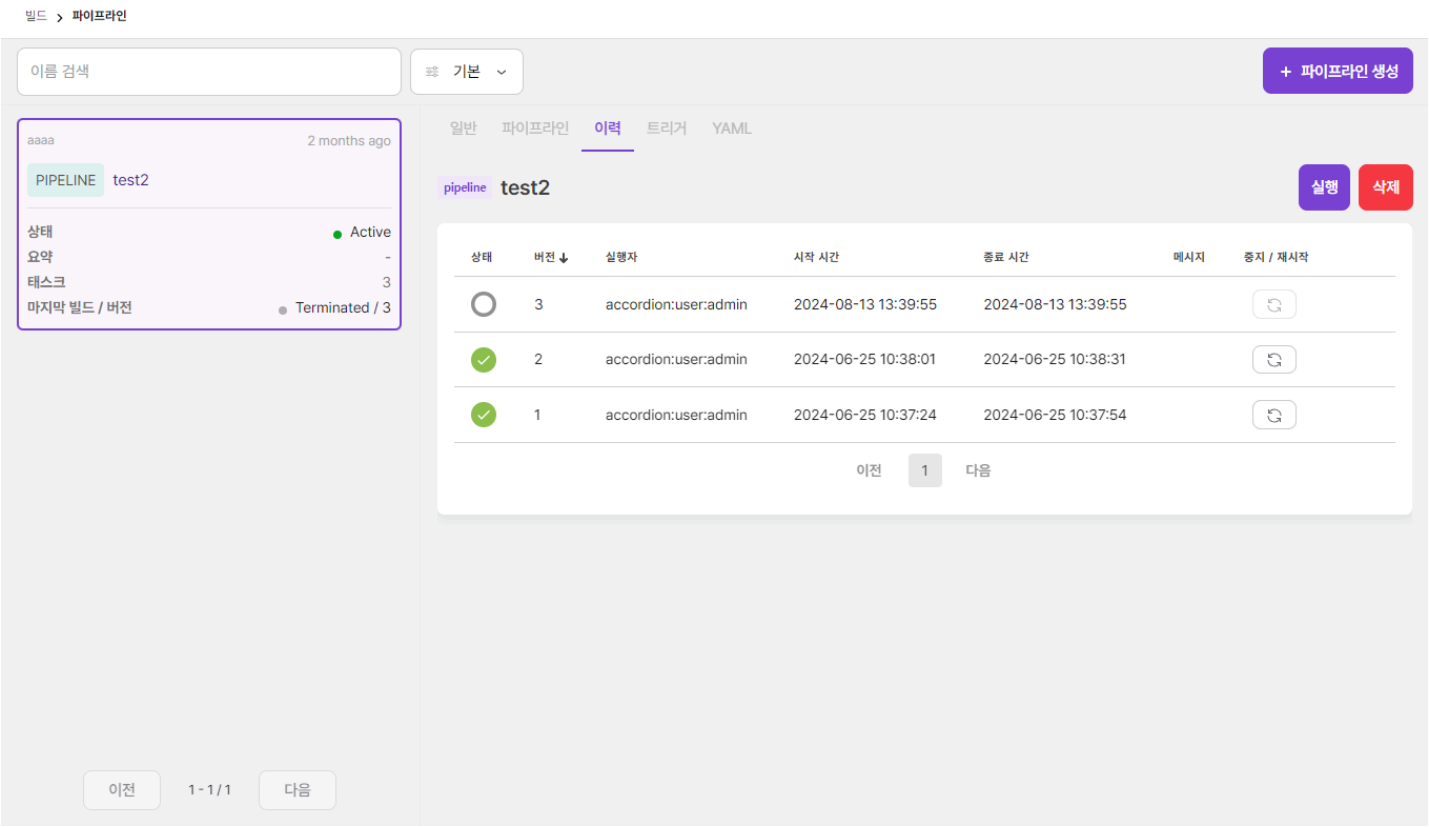
빌드보관 정책은 background 와 buildRequest 삭제 이벤트 발생을 기준으로 각각 동작한다.
count 또는 period annotation 이 존재하는 경우 삭제가 진행된다.
정상적으로 삭제되는 경우 카탈로그는 buildStatus.lastDeletedVersion 에 등록되며,
파이프라인은 status.lastDeletedVersion 에 등록된다.

### 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.

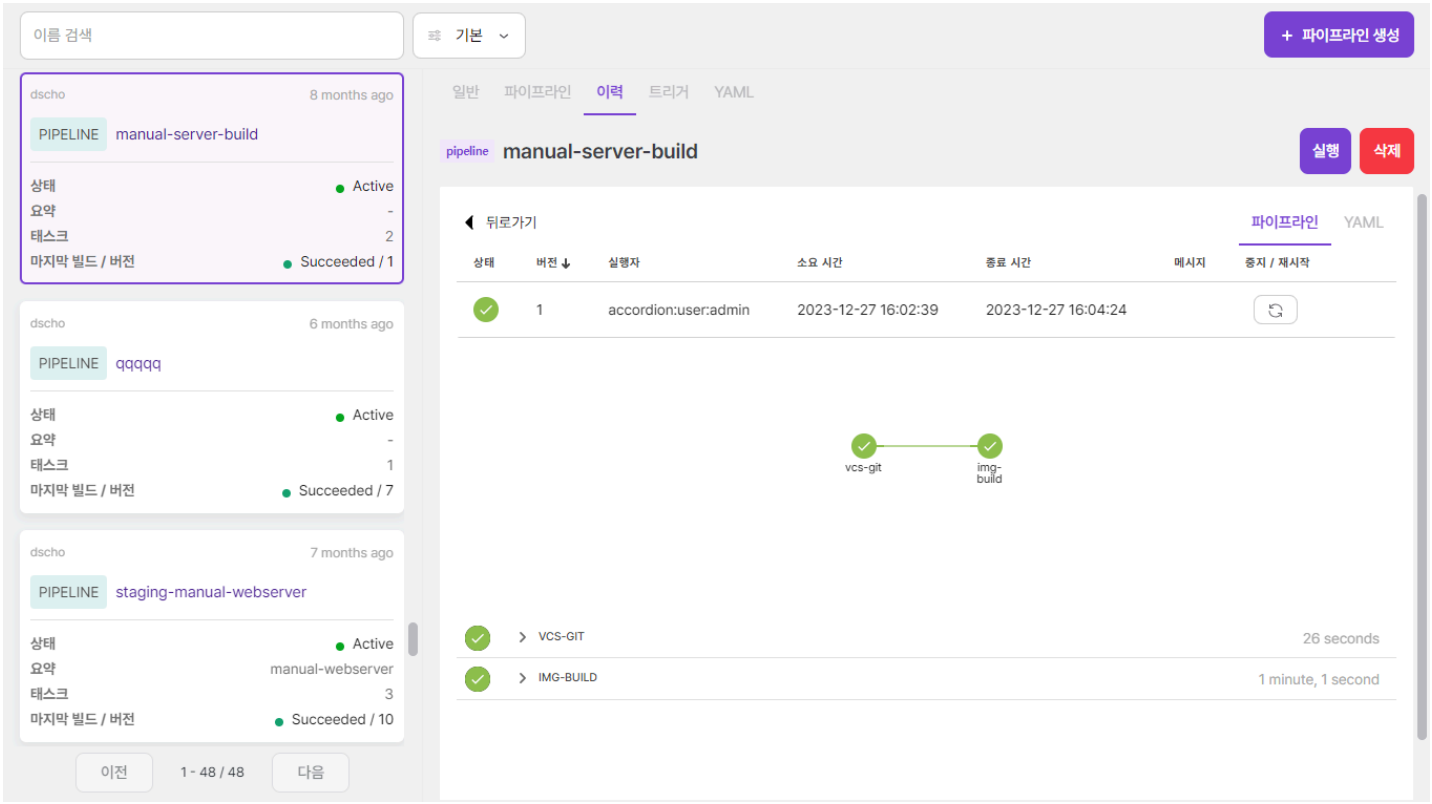



파이프라인 메뉴에서는 파이프라인을 실행한 빌드 이력을 확인할 수 있다. 이때 개별 빌드에 대한 상태 정보가 아이콘으로 표시되는데 각각이 의미하는 바는 다음과 같다.

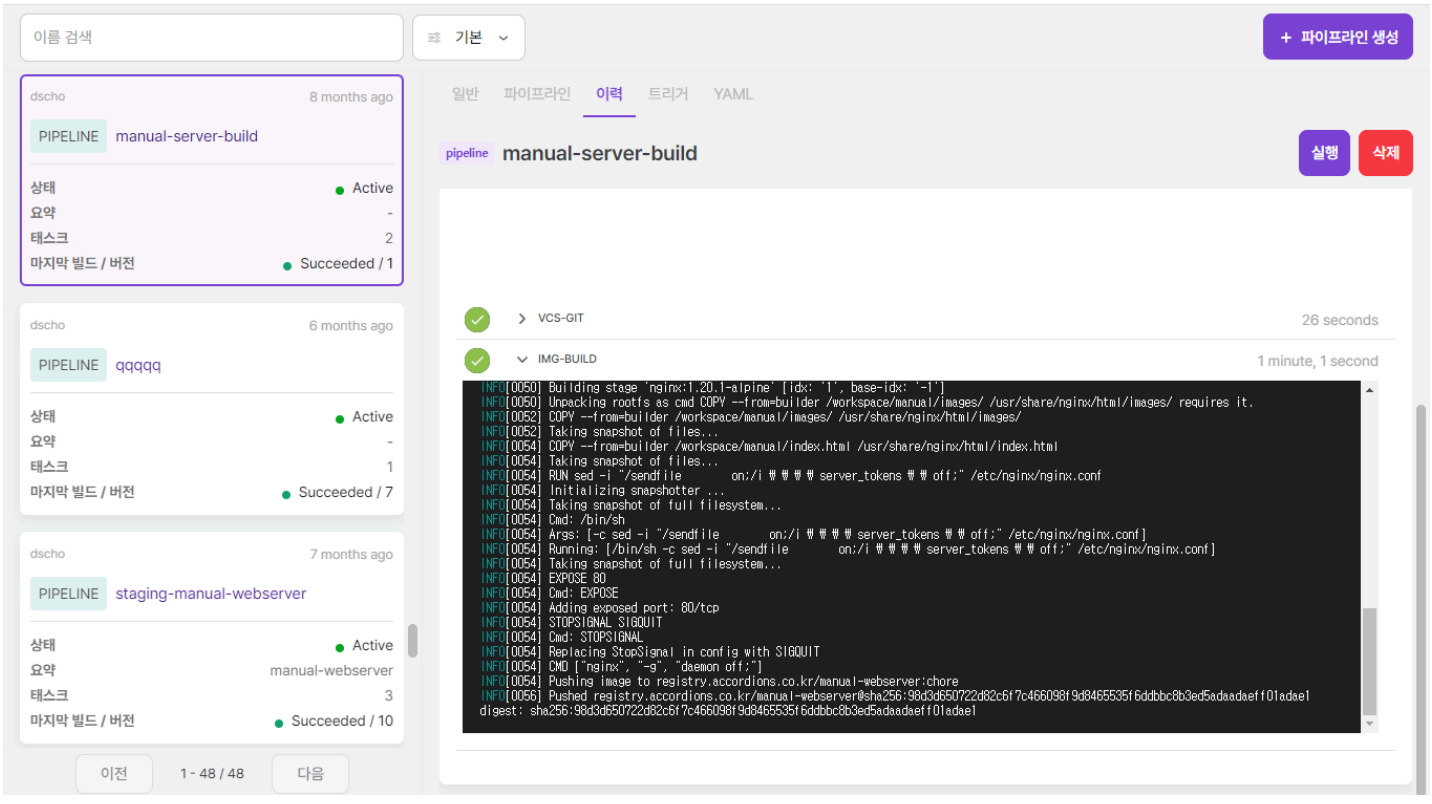
| 이미지 | 태스크 상태 |
| --- | --- |
| | 종료(Terminated): 사용자가 빌드를 중지한 상태 |
| | 성공(Succeeded) : 태스크가 수행하고 종료코드(exit code)가 0으로 완료된 상태 |
| | 실행 중(Running) : 태스크가 수행되고 있는 상태 |
| | 실패(Failed) : 태스크가 수행되었지만 종료코드가 0이 아닌 것으로 완료된 상태 |
| | 에러(Error) : 태스크가 실행되지 못하거나 비정상 종료된 상태<br><br>발생 예시<br><br>• 쿠버네티스 인프라적인 요인으로 태스크가 수행되지 않는 경우<br>• 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.

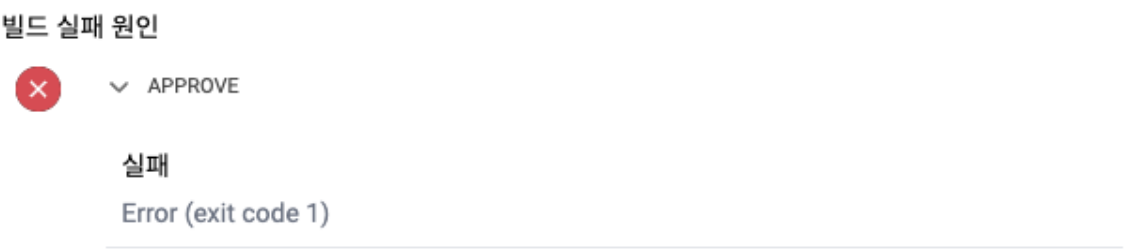



개별 빌드는 하나 이상의 태스크로 구성되며 태스크를 선택하면 해당 태스크에서 발생한 로그를 확인할 수 있다.



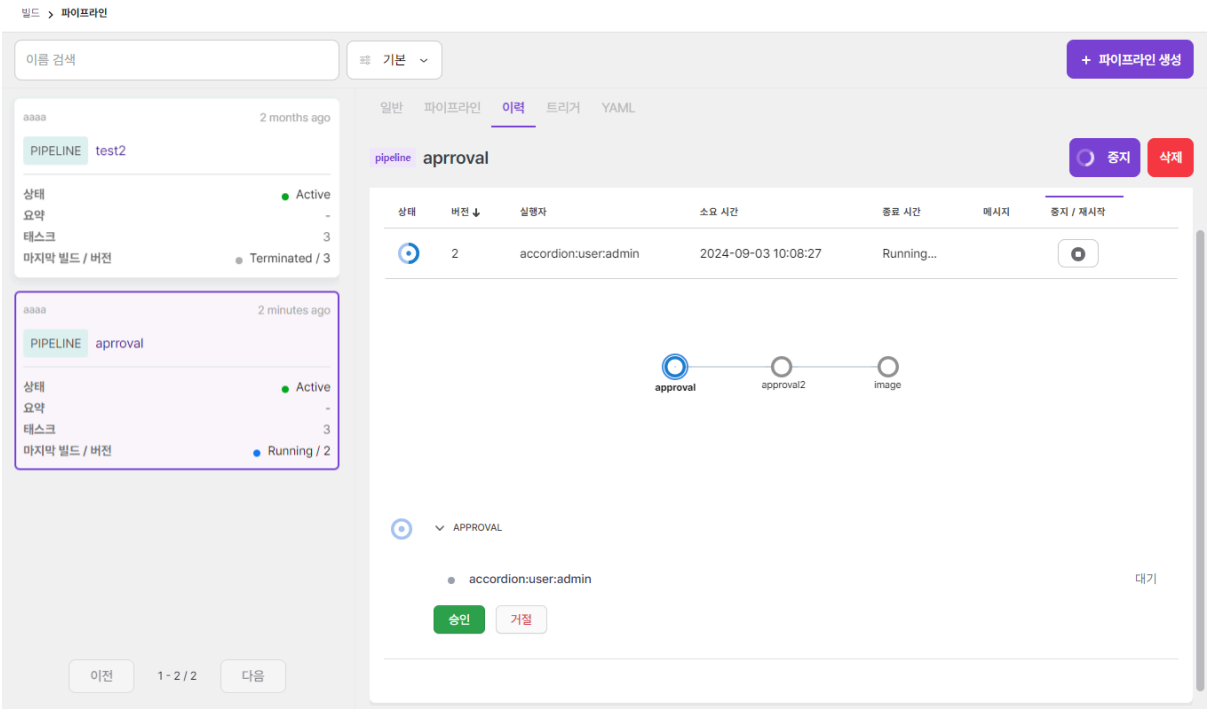
NOTE

실패한 태스크의 경우 실패한 원인을 확인할 수 있다.




NOTE

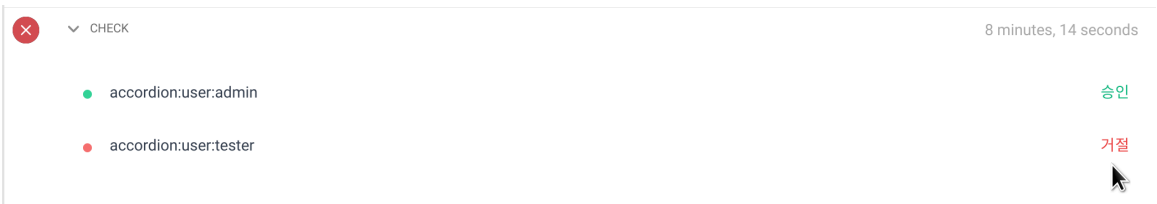
태스크에는 승인 태스크도 있다. 승인 태스크의 경우 태스크를 선택했을때 로그 정보가 제공되는 것이 아니라 승인 목록이 나타나고 사용자가 승인자 목록에 포함되어 있으면 승인 또는 거절이 가능하다.



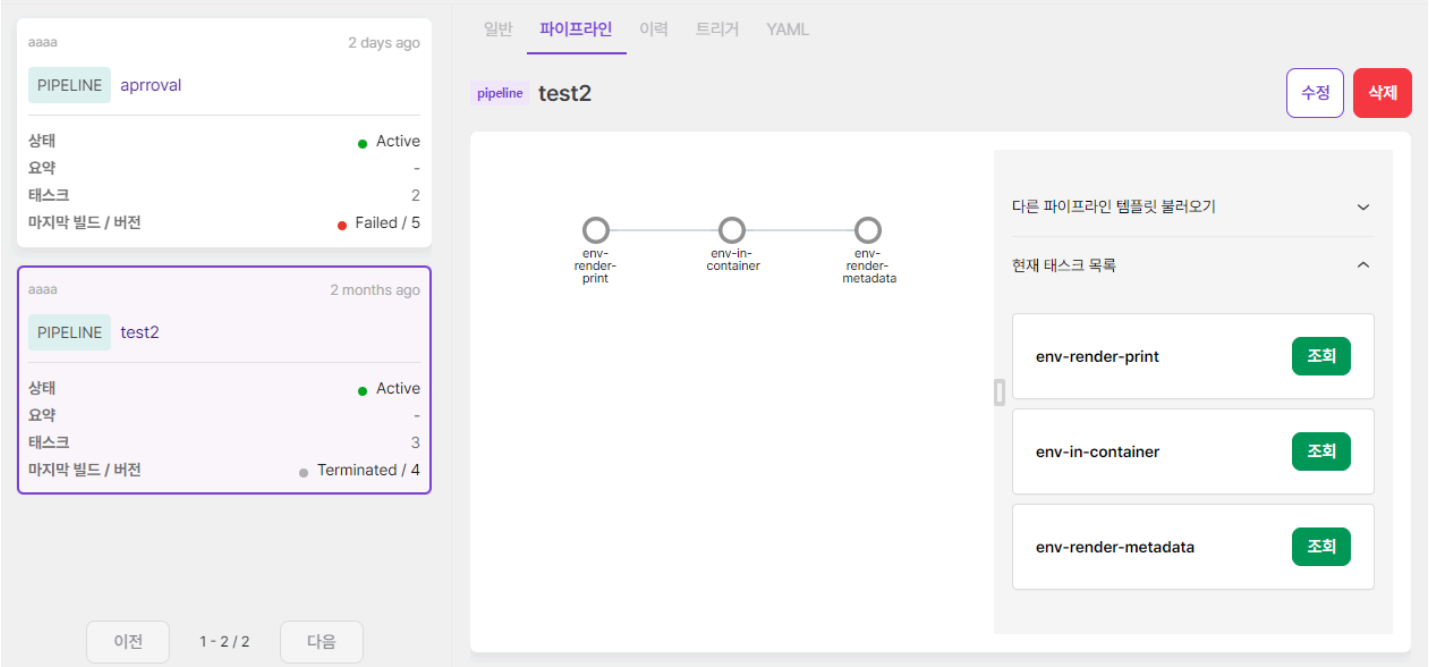

승인의 경우 모든 승인자가 승인해야 다음 태스크로 넘어가며 한명의 승인자라도 거절하면 파이프라인은 중지된다.

- 승인 성공

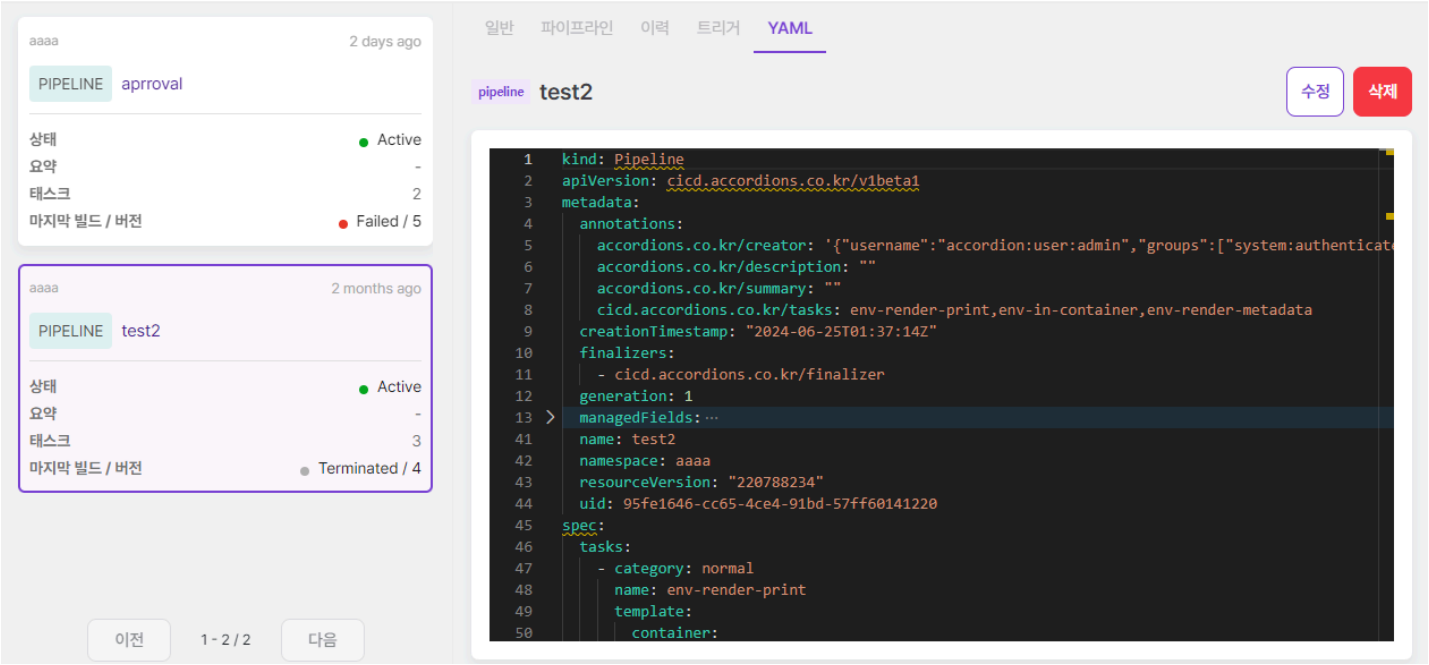
- 승인 실패



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 수정할 수 있다.



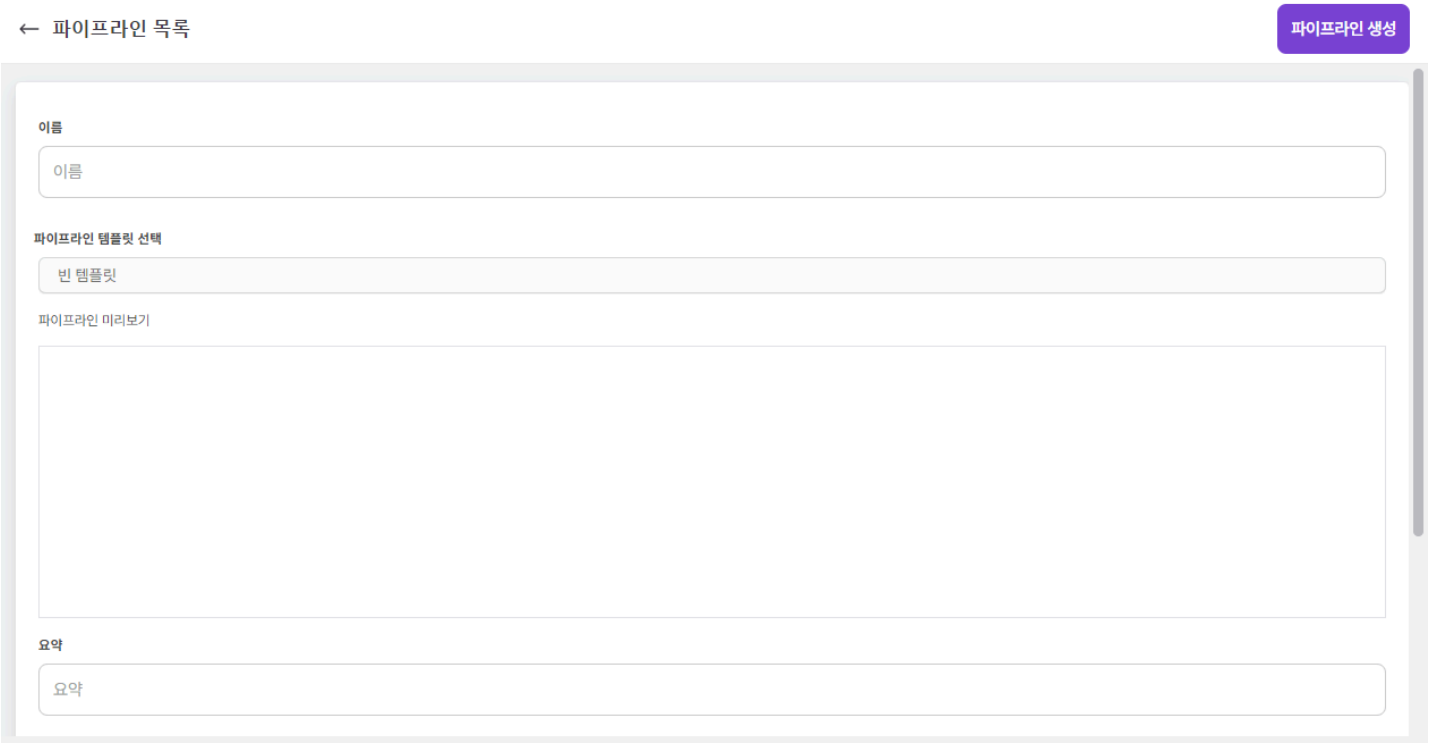

YAML 편집기로 보면 예약된 환경변수를 볼 수 있다. 파이프라인에서 예약된 환경변수는 다음과 같다.

| 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

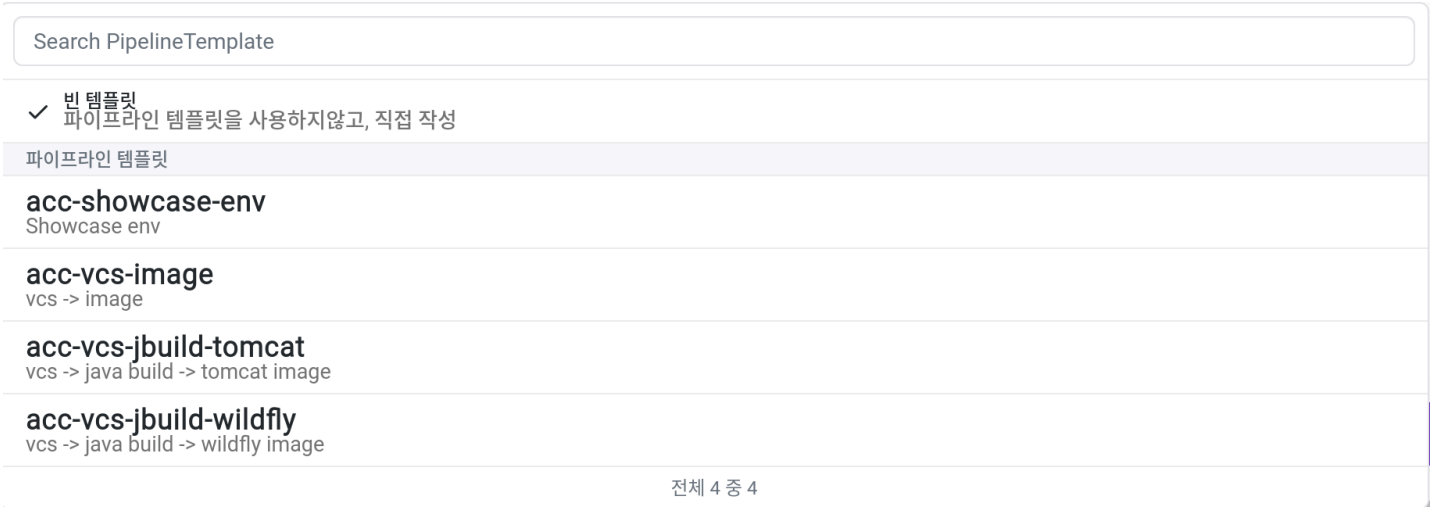
+ 파이프라인 생성 버튼을 선택하면 나타나는 화면에서 파이프라인 정보를 입력하여 생성할 수 있다. 생성 시에는 FORM 또는 YAML로 입력할 수 있다.




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 변경사항을 반영한다.

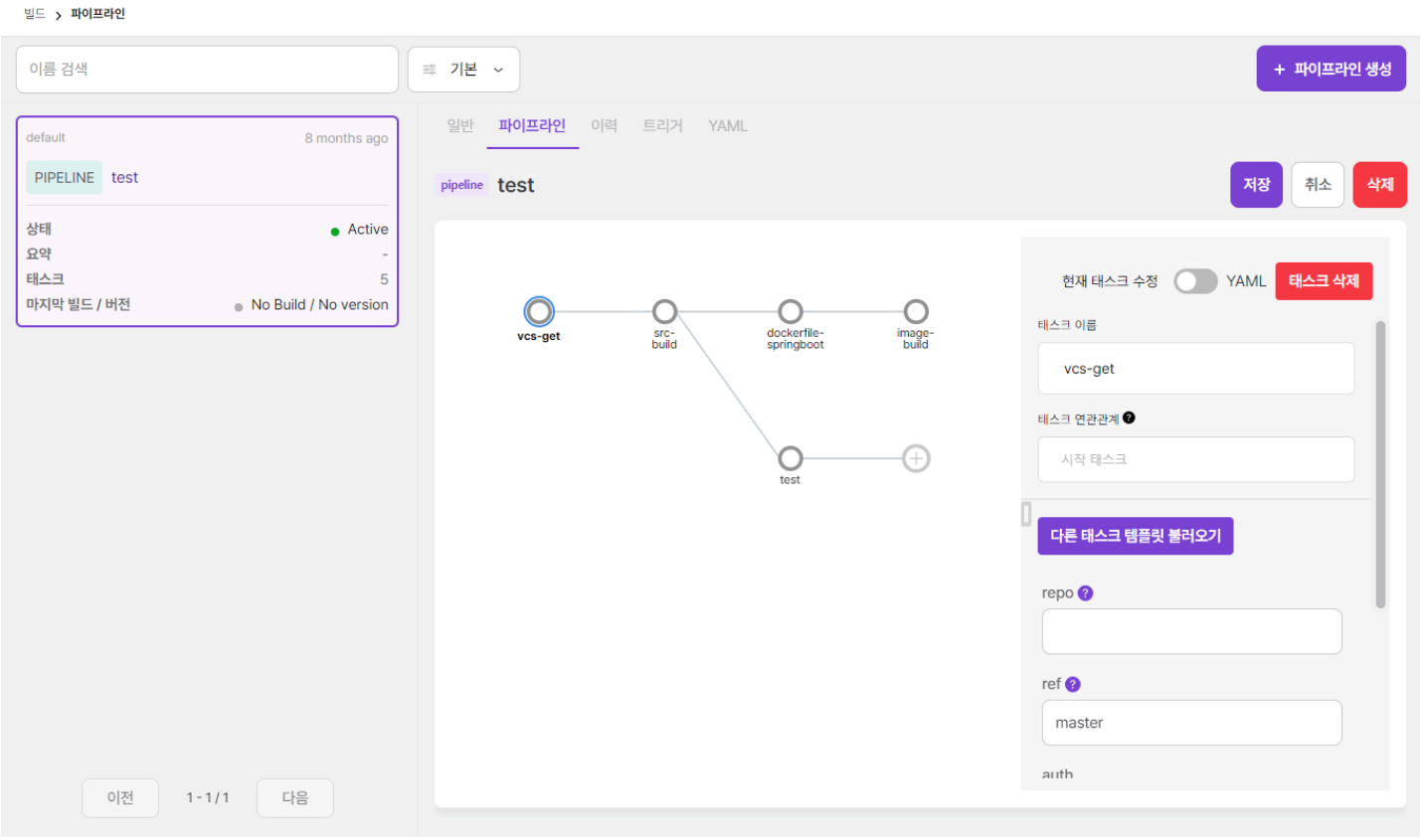



**TIP**

파이프라인 생성 시 빈 템플릿 을 선택하면 태스크가 없는 파이프라인을 생성할 수 있다. 이 경우 처음부터 파이프라인을 모두 작성할 수 있다.

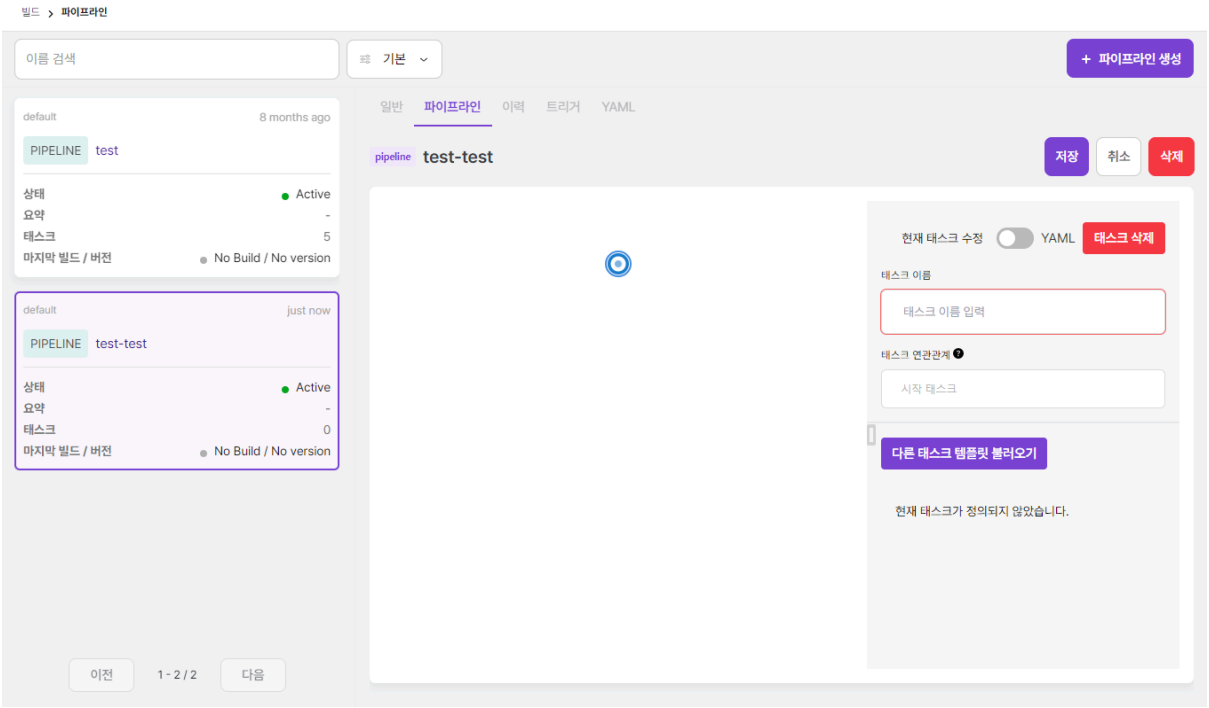
## 파이프라인 수정

파이프라인을 생성할 때 태스크에 대한 정보는 파이프라인 템플릿에서 가져와 구성한다. 파이프라인의 태스크에 대한 정보를 변경해야하는 경우 변경할 파이프라인을 찾아 `파이프라인` 탭의 `수정` 버튼을 선택한다.




**NOTE**

태스크가 없는 파이프라인의 경우에는 수정하여, 빈 태스크의 `+` 버튼을 선택하면 태스크 구성을 할 수 있다.

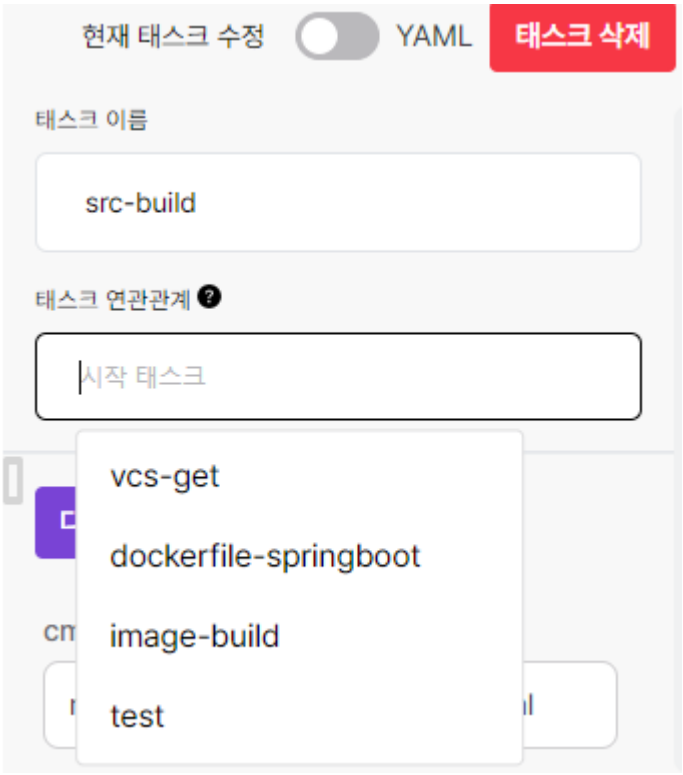



태스크에 대한 스펙을 작성할때에는 이름과 다른 태스크와의 연관관계를 설정하고 상세정보는 태스크 템플릿을 기반으로 작성한다.

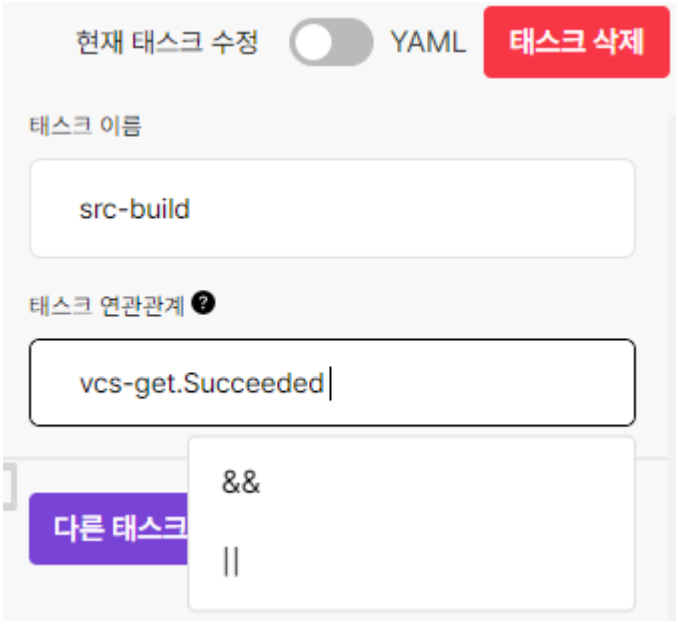
태스크의 이름은 쿠버네티스 이름 정책에 맞춰 작성하고 태스크의 연관관계는 해당 태스크을 수행하는 조건에 대한 정보를 입력한다. 이는 `{이름}.{상태}` 의 형식으로 입력할 수 있다. 상태에 입력할 수 있는 값은 다음과 같다.

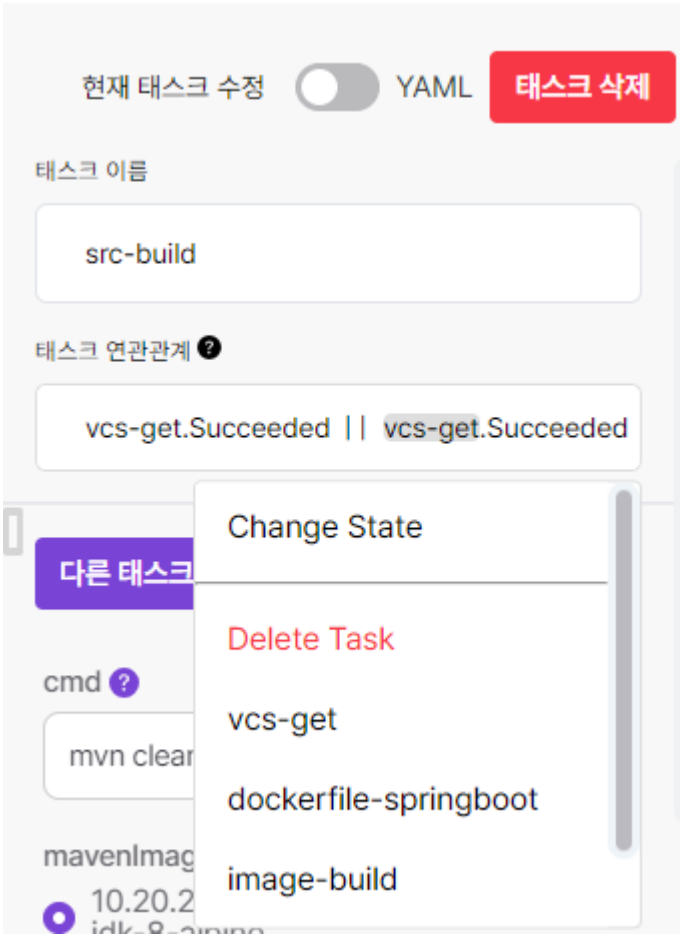
| 상태 | 설명 |
| --- | --- |
| Succeeded | 성공 |
| Failed | 실패 |
| Skipped | 생략 |
| Error | 에러 |

연관관계 작성 방법은 다음과 같다.

- 테스크 연결
  - 테스크 연관관계의 입력창을 선택한 후 테스크 설정
  - 

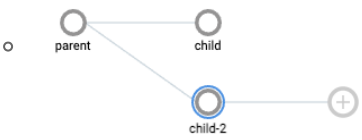
- 테스크 상태, 조건 추가
  - 순차적으로 테스크의 상태와 조건을 선택

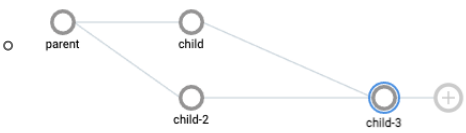
- 
현재 태스크 수정
YAML
태스크 삭제
태스크 이름
src-build
태스크 연관관계
vcs-get.Succeeded
&&
||
다른 태스크

- 테스크 삭제, 수정
  - 수정이 필요한 테스크, 상태를 클릭하여 내용 변경
  - 


**TIP**

연관관계 작성의 예는 다음과 같다.

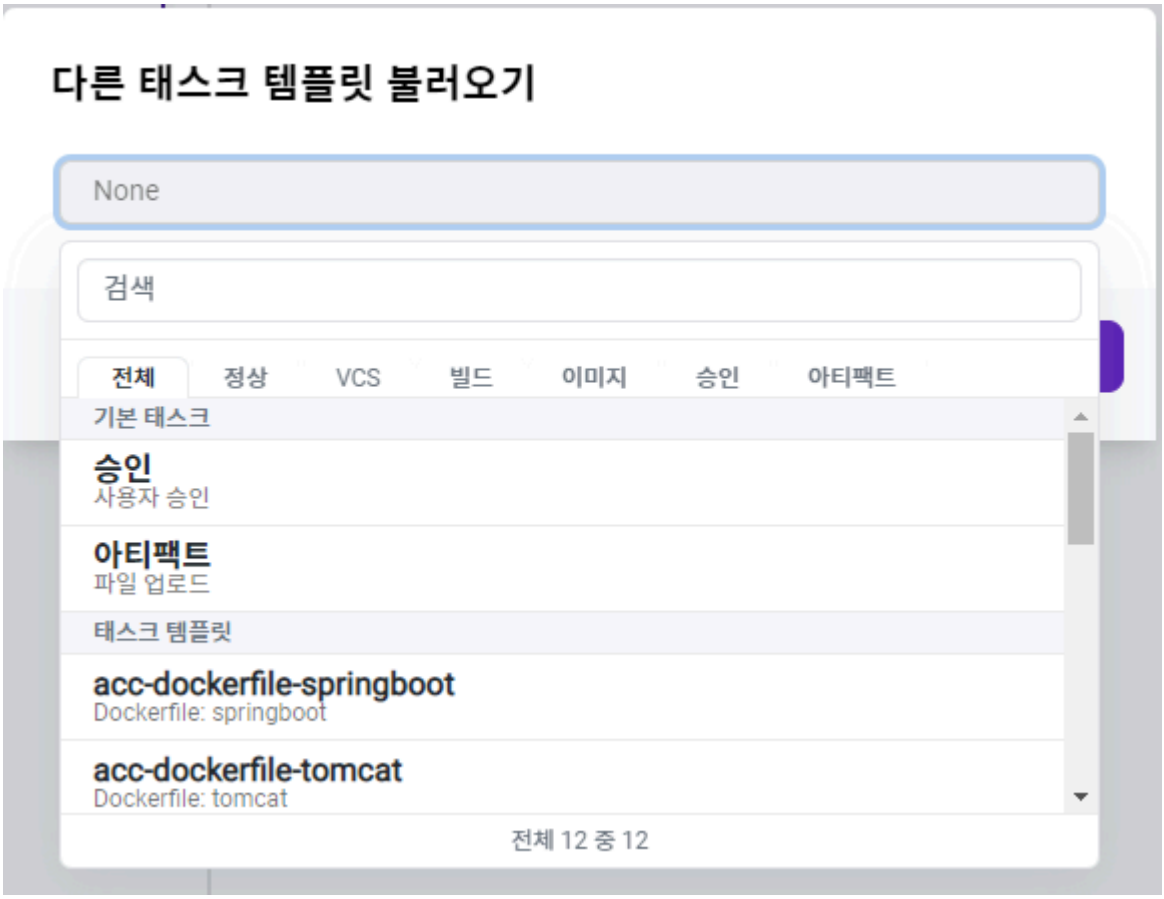
- 조건
  - parent 태스크 후 child 와 child-2 태스크를 수행한다.
  - child 태스크가 성공하고 child-2 태스크가 실패하면 child-3 태스크를 수행한다.
- 설정
  - 

  - 

**CAUTION**

파이프라인의 작업공간에 대한 프로비저닝 설정 구성 시, 접근 모드를 ReadWriteOnce로 구성할 경우 위 tip의 예시와 같이 연관관계를 child 와 child-2 태스크가 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.



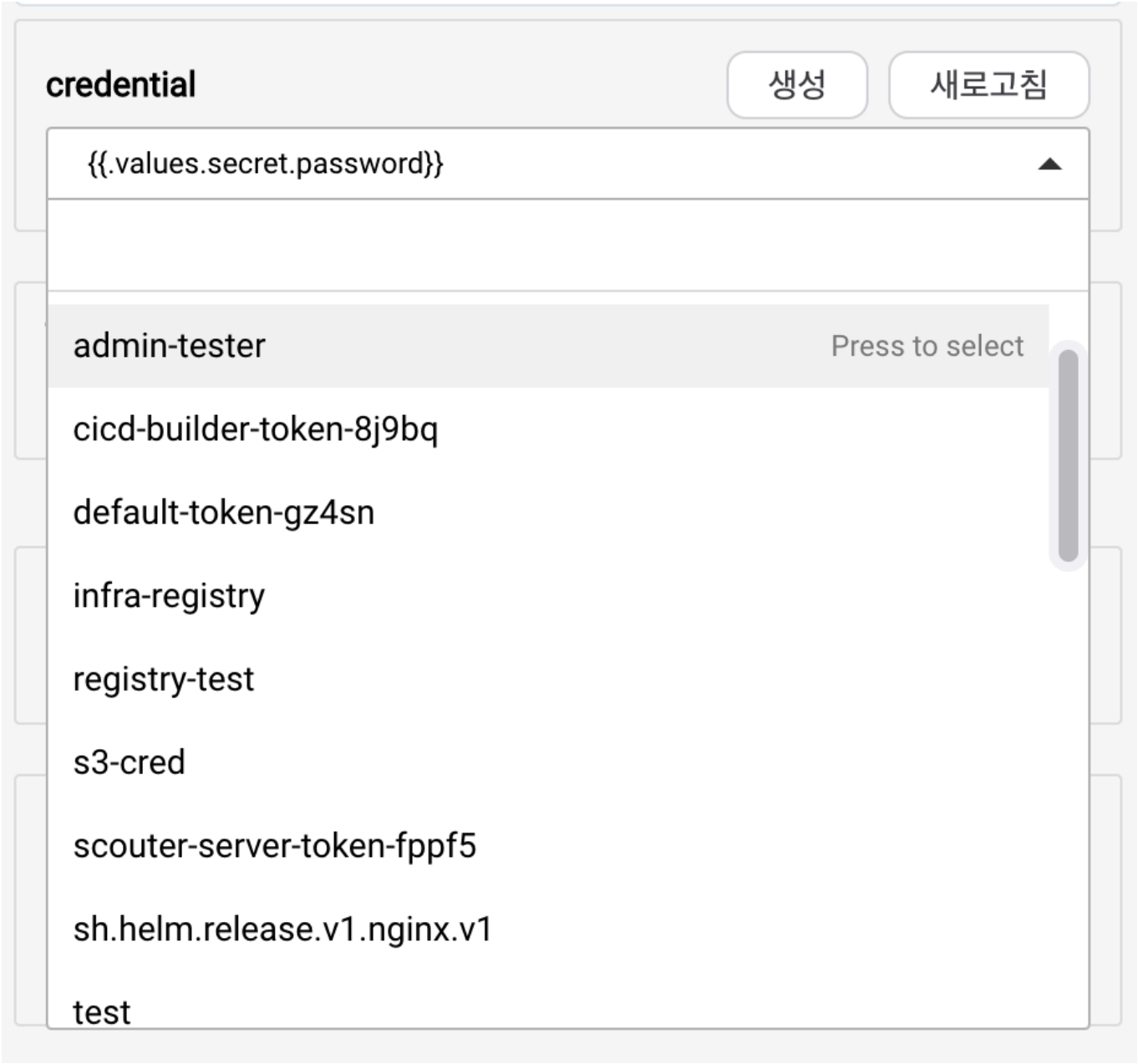

태스크 템플릿의 종류는 다음과 같다.

| 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<br>`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.

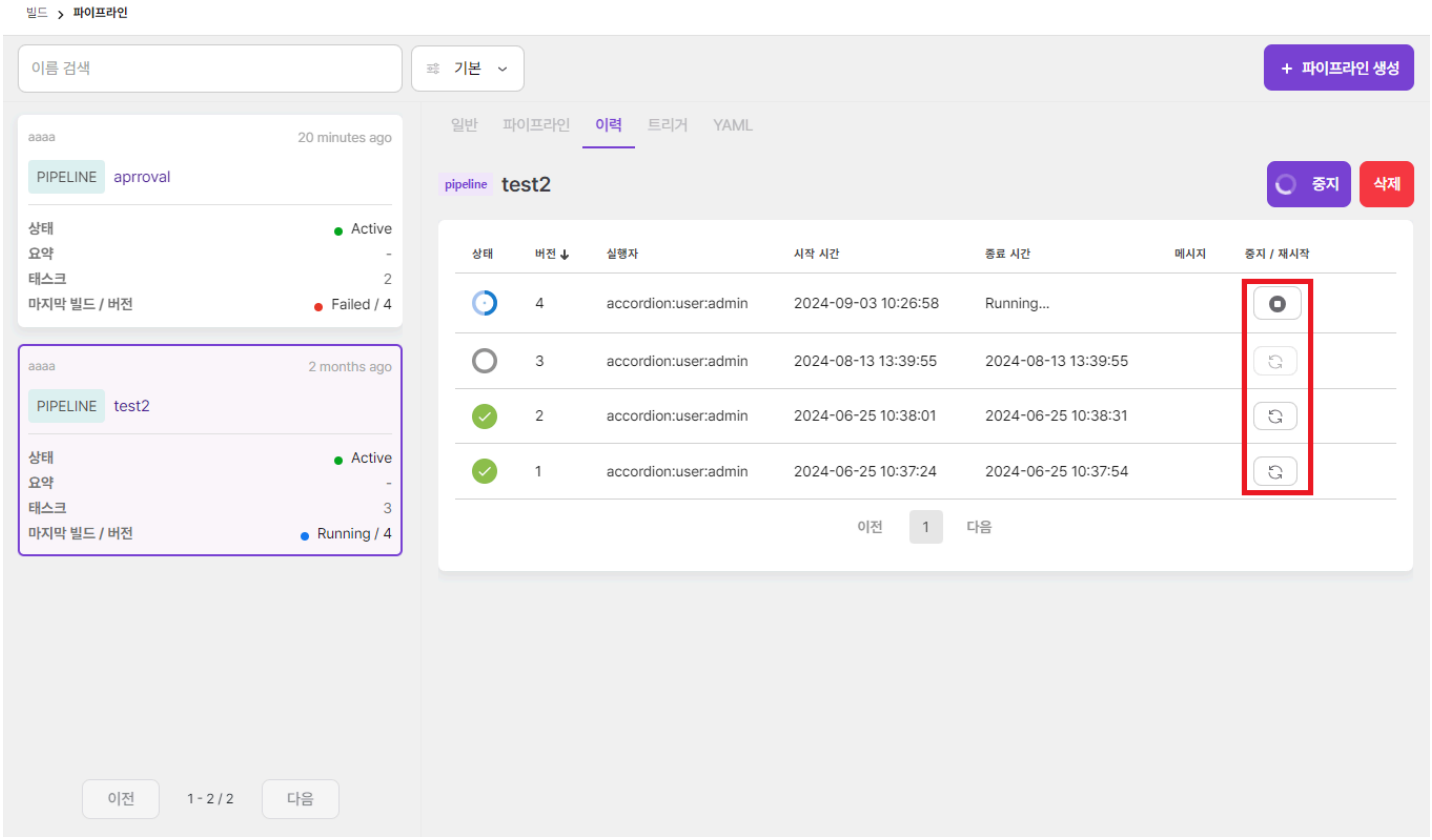
- 현재 사용자 권한에 따른 쿠버네티스 리소스 목록을 선택
- 환경변수를 입력




- 파이프라인 수정 시 쿠버네티스 리소스를 생성할 수 있는 경우는 `Secret`, `ConfigMap`, `PersistentVolumeClaim` 3가지 경우가 있다. 생성 화면은 각 메뉴별 생성 화면과 동일하다.

## 파이프라인 실행과 중지

이력 탭에서 실행 버튼을 선택하면 파이프라인의 빌드가 실행되는 모습을 확인할 수 있다.



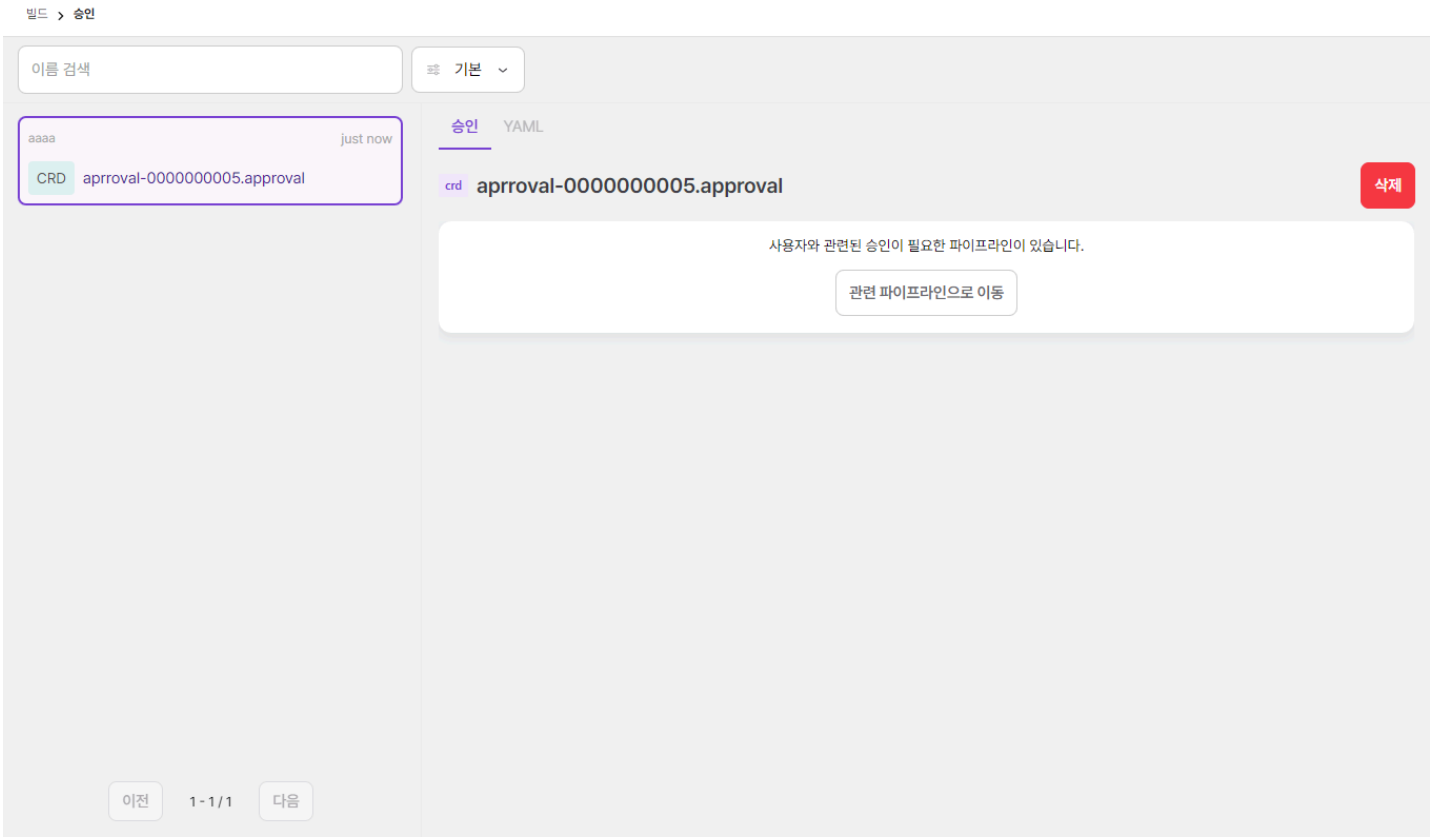

상단의 중지 버튼 또는 우측의 멈춤 아이콘 버튼을 선택하면 진행 중인 파이프라인의 빌드를 중지시킬 수 있다. 그 외에 빌드 내역 중 특정 빌드로 다시 배포하고 싶은 경우 해당 빌드 내역의 우측 버튼을 이용해 다시 배포도 가능하다. 이 경우 다시 실행하려는 빌드가 성공으로 완료되었을 때만 가능하다.

## 트리거

카탈로그의 트리거와 사용방법이 동일하기 때문에 해당 가이드를 참고 한다.

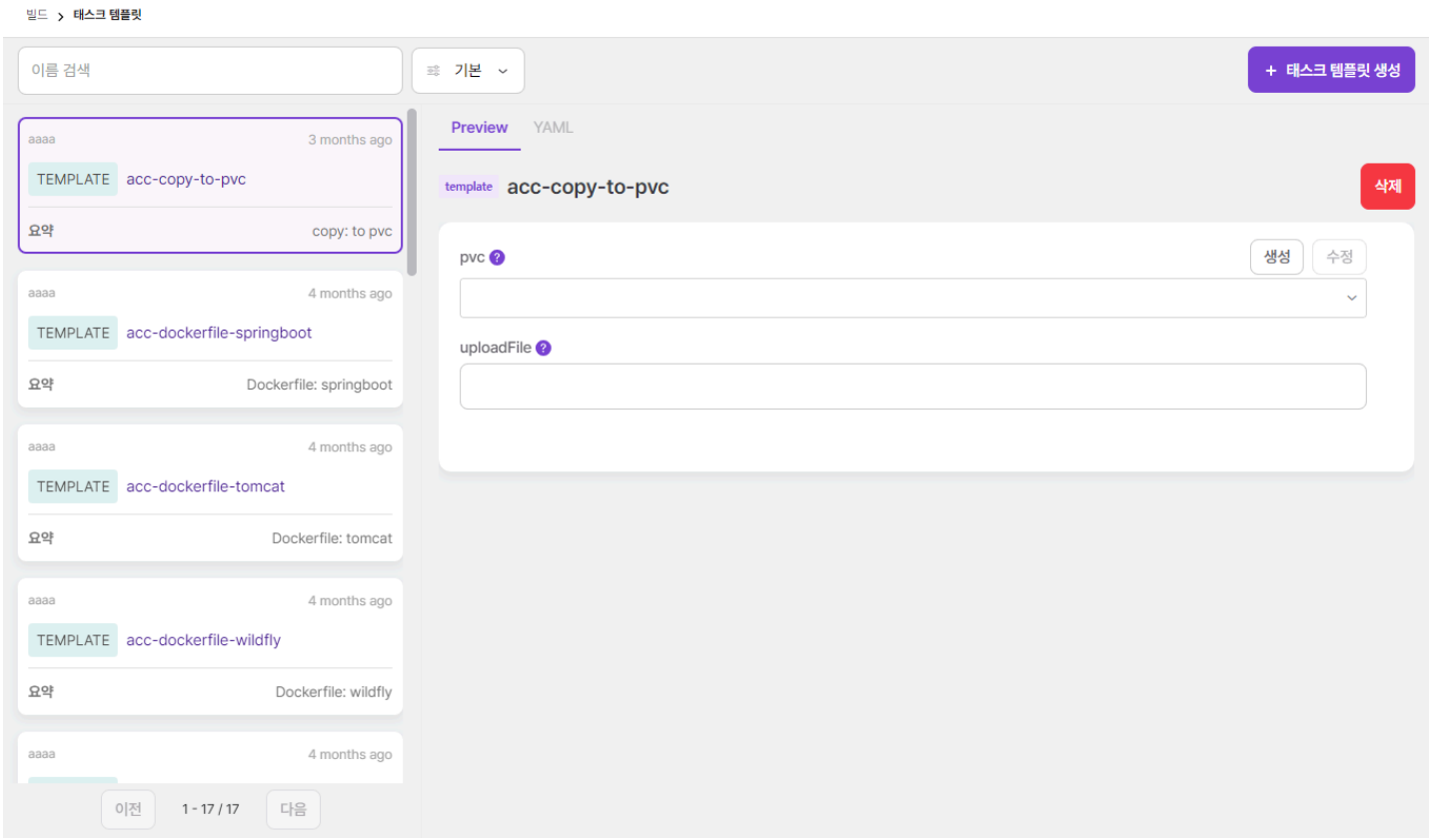
### 4.3.3.2. 승인

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 롤, 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.

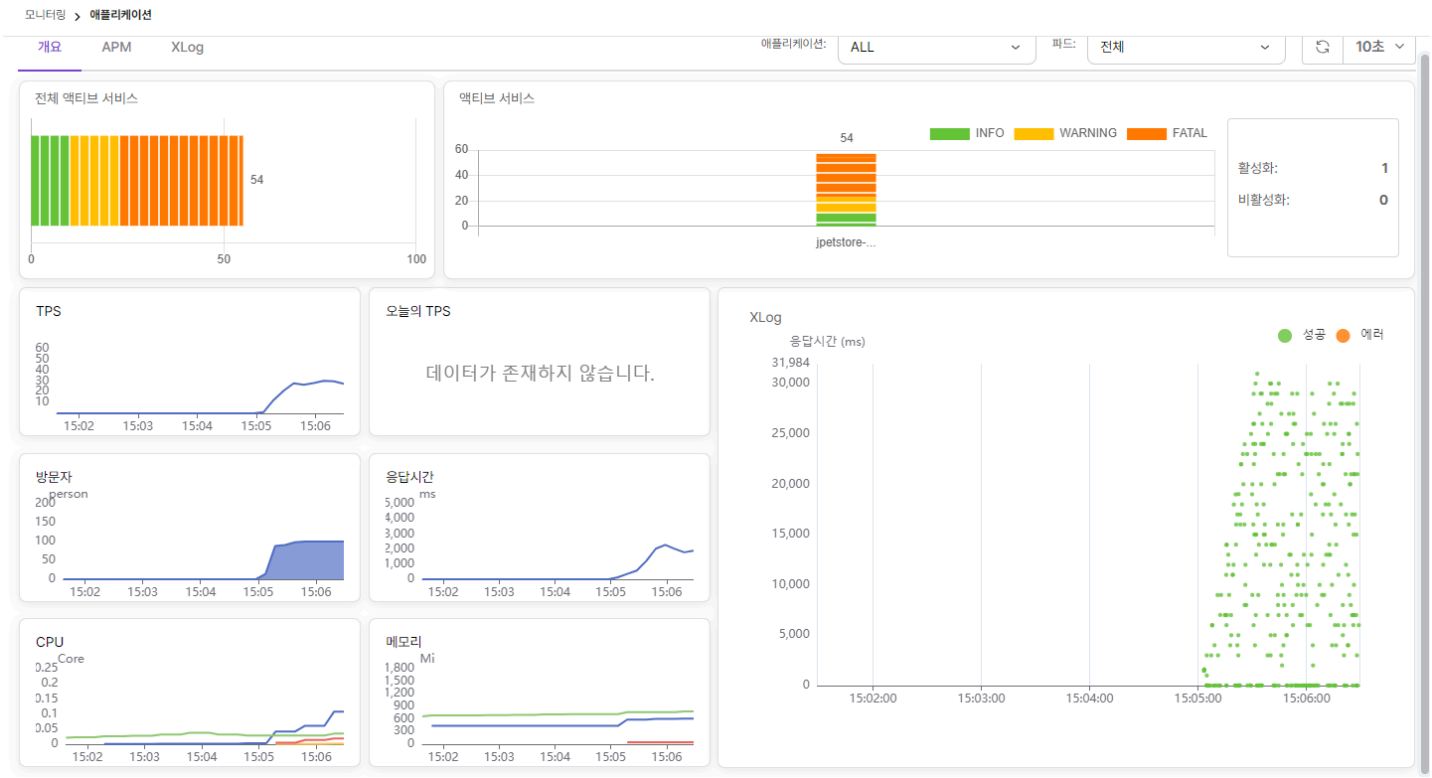
- 시스템: 클러스터/시스템
- 이벤트 로그: 클러스터/이벤트 로그
- 감사 로그: 클러스터/감사 로그
- 알림 로그: 클러스터/알림 로그

### 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 |



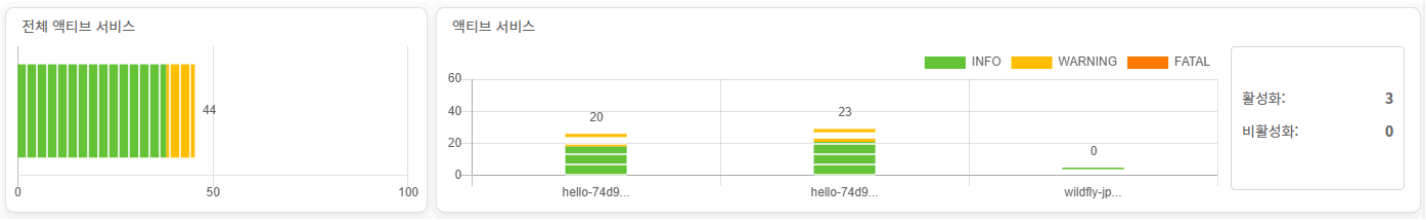

각 모니터링 항목에 대한 설명은 다음과 같다.

### 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.

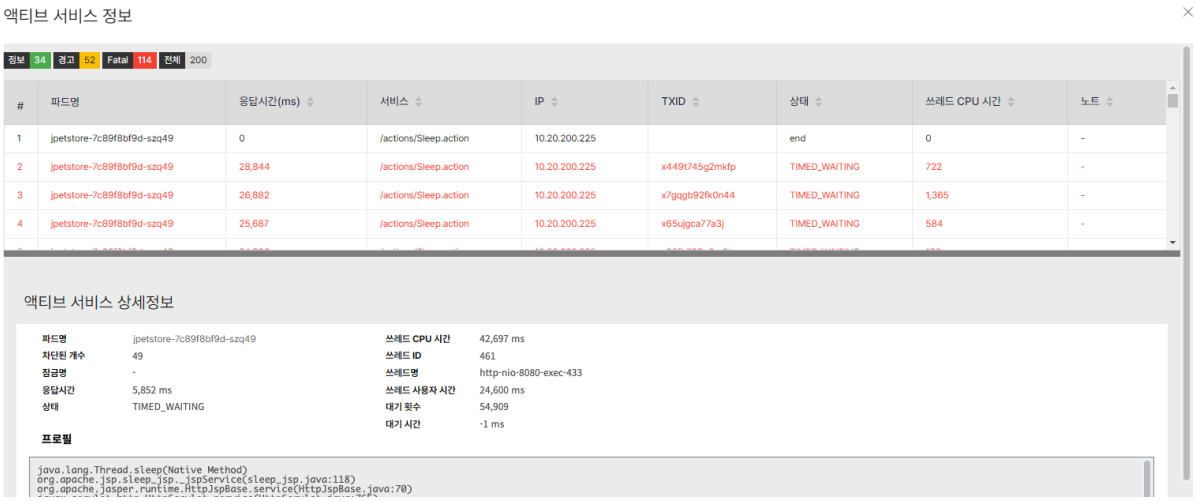
처리중인 리퀘스트들이 표시될 때, 해당 차트 선택하면 액티브 서비스 정보 모달이 나타나고 진행중인 리퀘스트에 대한 정보를 얻을 수 있다.




NOTE

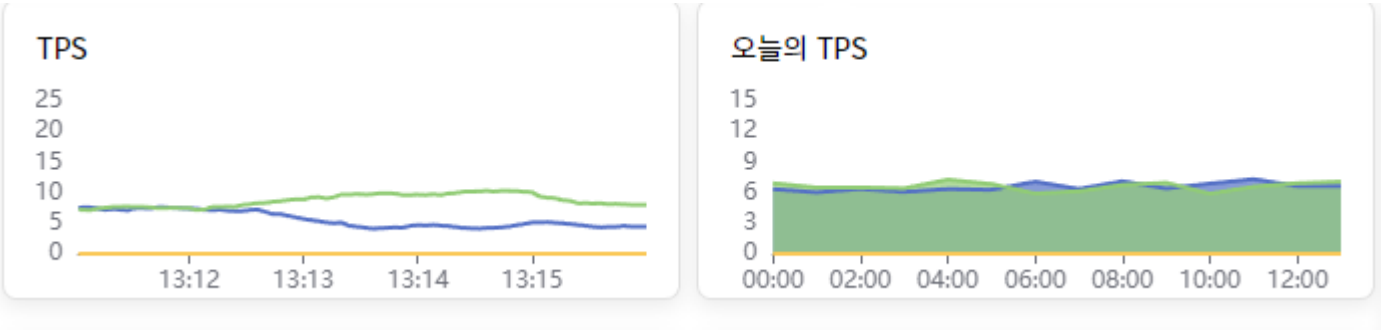
### 액티브 서비스 정보

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.



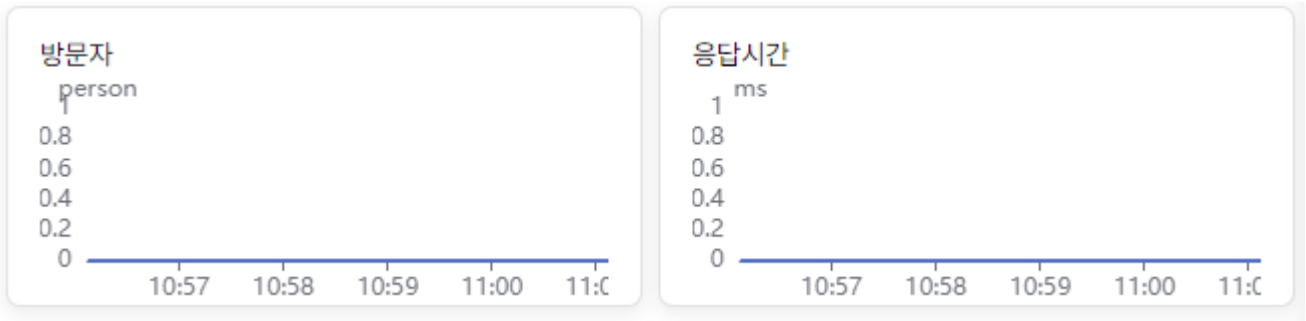
### TPS, 오늘의 TPS

TPS는 초당 처리량을 의미하고 오늘의 TPS는 당일 시간대별 TPS 의 평균을 의미한다.



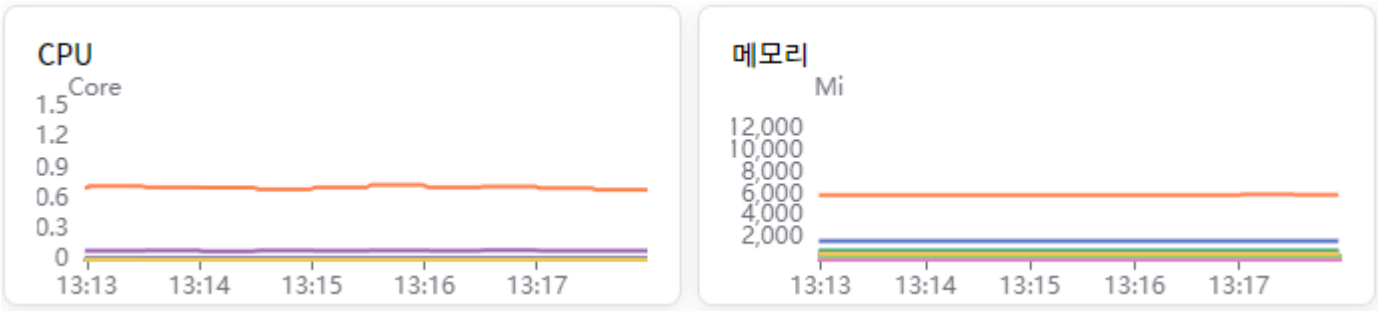

### 방문자, 응답시간

방문자는 5분간 방문한 유니크한 방문자 수를, 응답시간은 애플리케이션 응답시간을 의미한다.




### CPU, 메모리

CPU 와 메모리 사용량을 제공한다.

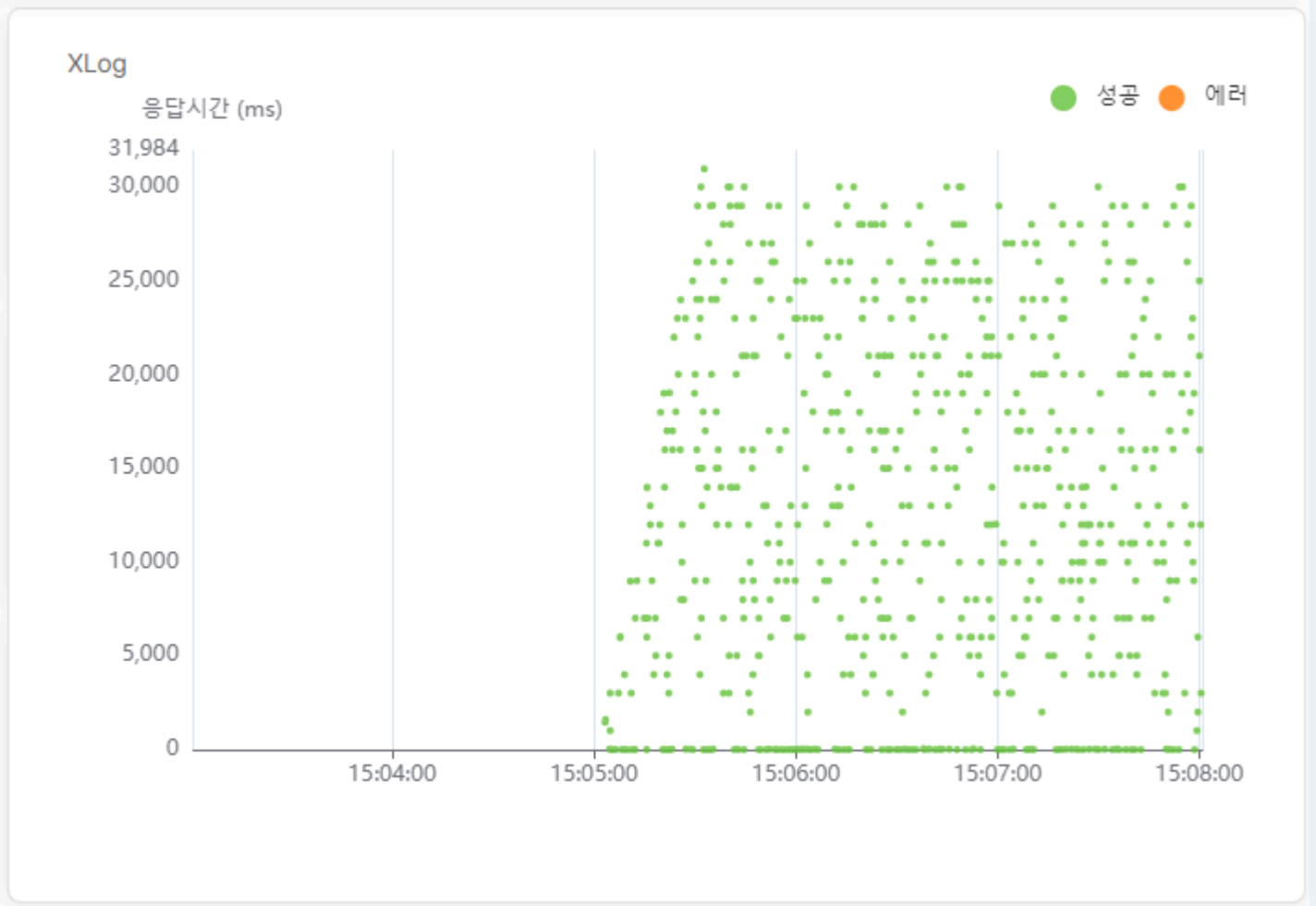



### XLog

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 Info 에서 해당 영역에서 처리된 요청들에 대한 상세 프로파일을 분석할 수 있다.

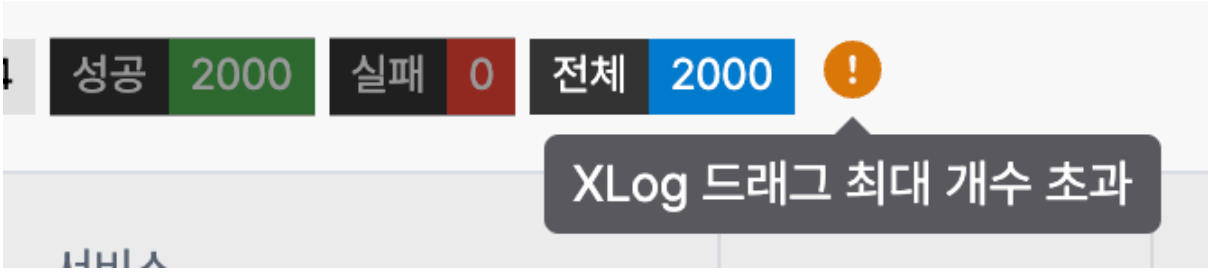
**NOTE** 콘솔에서 표시되는 XLog 최대 응답 크기의 기본값은 20MB 이다.




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.

**NOTE**

XLog 드래그 최대 개수는 기본 2000개로 설정되어 있으며, 이를 초과하면 드래그 범위를 줄이거나 설정을 변경해야 한다.



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 |

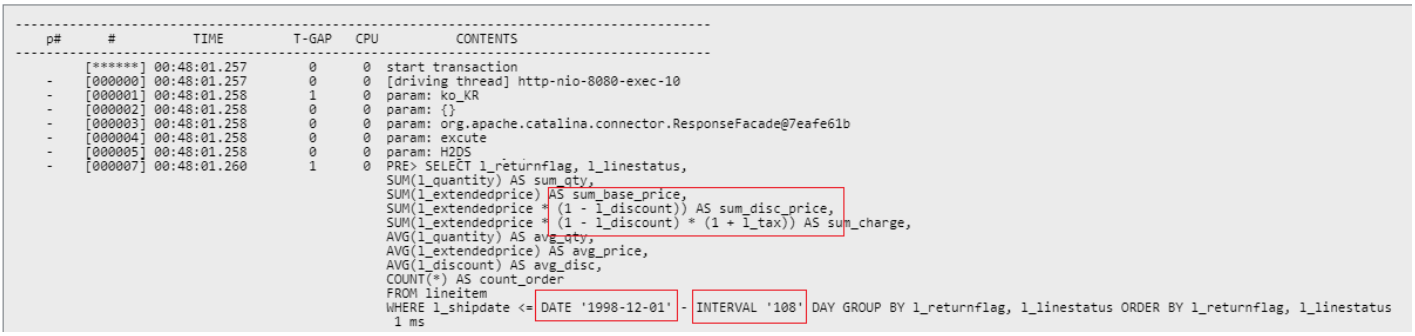
프로파일에서는 해당 메소드의 실행에 소요된 시간을 표시한다. 메소드들의 실행 관계는 트리 형태로 표현하며 이전 단계와 시간 차이를 T-GAP 으로 표시한다.

```
-----------------------------------------------------------------------------------------
    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.

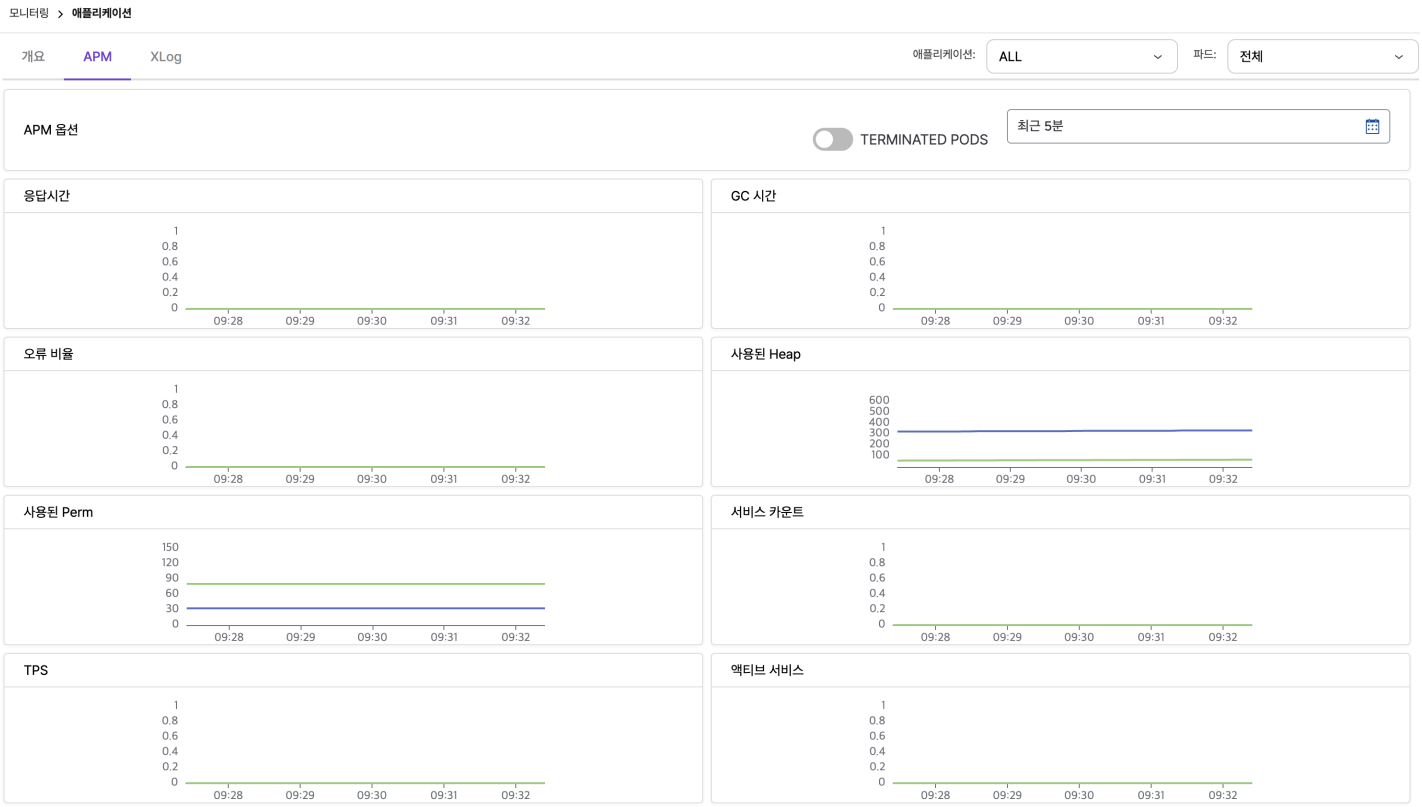


```
------------------------------------------------------------------------------------------
    p#      #          TIME         T-GAP   CPU          CONTENTS
------------------------------------------------------------------------------------------
        [******] 00:48:01.257        0      0  start 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 ORDER 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 |




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.