



# Red Hat Advanced Cluster Management for Kubernetes 2.10

다중 클러스터 글로벌 허브

다중 클러스터 글로벌 허브



# Red Hat Advanced Cluster Management for Kubernetes 2.10 다중 클러스터 글로벌 허브

---

다중 클러스터 글로벌 허브

## 법적 공지

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 초록

단일 허브로 여러 허브 클러스터를 관리할 수 있는 다중 클러스터 글로벌 허브에 대해 자세히 알아보려면 자세히 알아보십시오.

---

## 차례

<b>1장. 다중 클러스터 글로벌 허브 .....</b>	<b>3</b>
1.1. 다중 클러스터 글로벌 허브 아키텍처	3
1.2. 글로벌 허브 요구사항	4
1.3. 연결된 환경에 다중 클러스터 글로벌 허브 설치	7
1.4. 연결이 끊긴 환경에서 다중 클러스터 글로벌 허브 설치	8
1.5. 기존 구성 요소 통합	13
1.6. 기본 모드에서 관리 허브 클러스터 가져오기	16
1.7. GRAFANA 데이터 액세스	16
1.8. GRAFANA 경고 (기술 프리뷰)	17
1.9. CRON 작업 구성	20
1.10. 요약 프로세스 수동 실행	22
1.11. 다중 클러스터 글로벌 허브 백업 (기술 프리뷰)	23



# 1장. 다중 클러스터 글로벌 허브

다중 클러스터 글로벌 허브는 하나 이상의 허브 클러스터를 가져와서 단일 허브 클러스터에서 관리할 수 있는 구성 요소 집합입니다.

허브 클러스터를 관리 허브 클러스터로 가져온 후 다중 클러스터 글로벌 허브를 사용하여 모든 관리 허브 클러스터에서 다음 작업을 완료할 수 있습니다.

- 정책 규정 준수 상태 및 추세 보고
- 개요 페이지에서 모든 관리 허브 및 관리 클러스터를 인벤토리
- 불규칙한 정책 동작의 경우 감지 및 경고

다중 클러스터 글로벌 허브는 단일 허브 클러스터가 대규모 환경에서 다수의 클러스터를 관리할 수 없는 경우 유용합니다. 이 경우 클러스터를 더 작은 클러스터 그룹으로 나누고 각 그룹에 대한 허브 클러스터를 구성합니다.

해당 hub 클러스터에서 관리하는 관리 클러스터의 여러 허브 클러스터에서 데이터를 보는 것은 쉽지 않은 경우가 많습니다. 다중 클러스터 글로벌 허브를 사용하면 여러 허브 클러스터를 관리 허브 클러스터로 설계하여 여러 허브의 정보를 더 쉽게 볼 수 있습니다. 다중 클러스터 글로벌 허브 클러스터는 다른 허브 클러스터를 관리하고 관리 허브 클러스터에서 요약된 정보를 수집합니다.

**참고:** 다중 클러스터 글로벌 허브에는 Observability를 사용할 수 없습니다. 클러스터에 다중 클러스터 글로벌 허브를 설치하기 전에 다중 클러스터 관찰 기능 기능을 활성화한 경우 다중 클러스터 관찰 기능을 수동으로 비활성화합니다.

다중 클러스터 글로벌 허브를 사용하는 방법에 대한 자세한 내용은 다음 섹션을 참조하십시오.

- [다중 클러스터 글로벌 허브 아키텍처](#)
- [Global Hub 요구 사항](#)
- [연결된 환경에서 다중 클러스터 글로벌 허브 설치](#)
- [연결이 끊긴 환경에서 다중 클러스터 글로벌 허브 설치](#)
- [기존 구성 요소 통합](#)
- [기본 모드에서 관리 허브 클러스터 가져오기](#)
- [Grafana 데이터 액세스](#)
- [Grafana 경고 \(기술 프리뷰\)](#)
- [cron 작업 구성](#)
- [요약 프로세스 수동 실행](#)
- [다중 클러스터 글로벌 허브 백업\(기술 프리뷰\)](#)

## 1.1. 다중 클러스터 글로벌 허브 아키텍처

다중 클러스터 글로벌 허브는 허브 클러스터에 액세스하고 관리하는 데 사용되는 다음 구성 요소로 구성됩니다.

- 관리 톨과 콘솔이 실행되는 *글로벌 허브 클러스터* 라는 서버 구성 요소

- 글로벌 허브 클러스터에서 *관리할 수 있는 관리 허브* 라는 Red Hat Advanced Cluster Management에 설치된 클라이언트 구성 요소입니다. 관리 허브는 다른 클러스터도 관리합니다. 다중 클러스터 글로벌 허브 클러스터에 전용 클러스터를 사용할 필요가 없습니다.

다음 섹션에서 아키텍처에 대해 자세히 알아보십시오.

다음 고급 다중 클러스터 용어 및 구성 요소를 참조하십시오.

- [다중 클러스터 글로벌 허브 Operator](#)
- [다중 클러스터 글로벌 허브 관리자](#)
- [다중 클러스터 글로벌 허브 에이전트](#)
- [다중 클러스터 글로벌 허브 시각화](#)

### 1.1.1. 다중 클러스터 글로벌 허브 Operator

다중 클러스터 글로벌 허브 Operator에는 다중 클러스터 글로벌 허브의 구성 요소가 포함되어 있습니다. Operator는 글로벌 다중 클러스터 관리에 필요한 모든 구성 요소를 배포합니다. 구성 요소에는 **multicluster-global-hub-manager**, **multicluster-global-hub-grafana** 가 포함되며 관리 허브 클러스터의 다중 클러스터 글로벌 허브 클러스터 및 **multicluster-global-hub-agent** 에 **Kafka** 및 **PostgreSQL** 이 제공됩니다.

또한 Operator는 **매니페스트 작업** 사용자 정의 리소스를 활용하여 관리형 클러스터에 Red Hat Advanced Cluster Management for Kubernetes Operator를 배포합니다. Red Hat Advanced Cluster Management Operator를 관리 클러스터에 배포한 후 관리 클러스터는 표준 Red Hat Advanced Cluster Management 클러스터가 됩니다. 이 hub 클러스터는 이제 관리 허브 클러스터입니다.

### 1.1.2. 다중 클러스터 글로벌 허브 관리자

다중 클러스터 글로벌 허브 관리자는 데이터를 **postgreSQL** 데이터베이스에 저장하는 데 사용됩니다. 데이터는 Kafka 전송에서 가져온 것입니다. 관리자는 또한 데이터를 Kafka 전송에 게시하므로 관리 허브 클러스터의 데이터와 동기화할 수 있습니다.

### 1.1.3. 다중 클러스터 글로벌 허브 에이전트

다중 클러스터 글로벌 허브 에이전트는 관리 허브 클러스터에서 실행됩니다. 다중 클러스터 글로벌 허브 클러스터와 관리 허브 클러스터 간의 데이터를 동기화합니다. 예를 들어 에이전트는 관리 허브 클러스터의 관리 클러스터 정보를 다중 클러스터 글로벌 허브 클러스터로 동기화하고 다중 클러스터 글로벌 허브 클러스터에서 관리 허브 클러스터로 정책 또는 애플리케이션을 동기화합니다.

### 1.1.4. 다중 클러스터 글로벌 허브 시각화

Grafana는 다중 클러스터 글로벌 허브 시각화의 기본 서비스로 다중 클러스터 글로벌 허브 클러스터에서 실행됩니다. Global Hub Manager에서 수집한 PostgreSQL 데이터는 기본 DataSource입니다. **multicluster-global-hub-grafana** 라는 경로를 사용하여 서비스를 노출하면 콘솔에 액세스하여 다중 클러스터 글로벌 허브 Grafana 대시보드에 액세스할 수 있습니다.

## 1.2. 글로벌 허브 요구사항

설치 및 네트워킹에 필요한 것은 물론 지원되는 구성 요소 및 환경에 대해 알아보십시오.

- [일반 요구 사항](#)



- 네트워킹 요구사항
- 지원되는 구성 요소

### 1.2.1. 일반 요구 사항

글로벌 허브를 설치하려면 다음 요구 사항이 필요합니다.

**필수 액세스:** 클러스터 관리자

**OpenShift Container Platform Dedicated 환경에 필요한 액세스 권한:** **cluster-admin** 권한이 있어야 합니다. 기본적으로 **dedicated-admin** 역할에는 OpenShift Container Platform Dedicated 환경에서 네임스페이스를 생성하는 데 필요한 권한이 없습니다.

- Red Hat Advanced Cluster Management for Kubernetes를 설치하고 구성해야 합니다. [Red Hat Advanced Cluster Management](#)에 대해 자세히 알아보십시오.

### 1.2.2. 네트워킹 요구사항

다음 네트워킹 요구 사항을 참조하십시오.

- 관리형 허브는 Red Hat Advanced Cluster Management에서 다중 클러스터 글로벌 허브의 관리형 클러스터이기도 합니다. Red Hat Advanced Cluster Management의 네트워크 구성이 필요합니다. Red Hat Advanced Cluster Management 네트워킹 세부 정보는 네트워킹을 참조하십시오. [https://access.redhat.com/documentation/en-us/red\\_hat\\_advanced\\_cluster\\_management\\_for\\_kubernetes/2.9/html/networking/networking](https://access.redhat.com/documentation/en-us/red_hat_advanced_cluster_management_for_kubernetes/2.9/html/networking/networking)
- 다음 표에는 글로벌 허브 네트워크 정보가 나열되어 있습니다.

방향	프로토콜	연결	포트(지정된 경우)	소스 주소	대상 주소
사용자의 브라우저에서 인바운드	HTTPS	사용자가 Grafana 대시보드에 액세스해야 합니다.	443	사용자의 브라우저	Grafana 경로의 IP 주소
Kafka 클러스터의 아웃바운드	HTTPS	글로벌 허브 관리자는 Kafka 클러스터에서 데이터를 가져와야 합니다.	443	multicluster-global-hub-manager-xxx pod	Kafka 경로 호스트

방향	프로토콜	연결	포트(지정된 경우)	소스 주소	대상 주소
PostgreSQL 데이터베이스의 아웃 바운드	HTTPS	글로벌 허브 관리자는 PostgreSQL 데이터베이스에 데이터를 유지해야 합니다.	443	multicluster-global-hub-manager-xxx pod	PostgreSQL 데이터베이스의 IP 주소

- 다음 표에는 Managed hub 네트워크 정보가 나열되어 있습니다.

방향	프로토콜	연결	포트(지정된 경우)	소스 주소	대상 주소
Kafka 클러스터의 아웃 바운드	HTTPS	글로벌 허브 에이전트는 Kafka 클러스터에 클러스터 정보 및 정책 정보를 동기화해야 합니다.	443	multicluster-global-hub-agent Pod	Kafka 경로 호스트

- 제품 문서에서 [Red Hat Advanced Cluster Management 클러스터 크기 조정 지침](#)을 확인할 수 있습니다.
- **선택 사항:** 미들웨어의 경우 다중 클러스터 글로벌 허브에는 Kafka, PostgreSQL 및 Grafana가 내장되어 있지만 자체 구성된 Kafka, PostgreSQL 및 Grafana를 사용할 수 있습니다. 자세한 내용은 [기존 구성 요소 통합](#)을 참조하십시오.

### 1.2.3. 지원되는 구성 요소

지원되는 플랫폼 및 구성 요소에 대해 알아보십시오.

- 통합 콘솔을 공유하므로 다중 클러스터 글로벌 허브 콘솔은 OpenShift Container Platform과 동일한 브라우저를 지원합니다. 지원되는 브라우저 및 버전에 대한 정보는 Red Hat OpenShift Container Platform 설명서에서 [웹 콘솔에 액세스](#)를 참조하십시오.
- 지원되는 다중 클러스터 글로벌 허브 클러스터에 사용 가능한 플랫폼은 다음 표에 표시되어 있습니다.

플랫폼	글로벌 허브 클러스터 지원	관리 허브 클러스터 지원
Red Hat Advanced Cluster Management 2.10 이상 2.10.x 릴리스	제공됨	제공됨

플랫폼	글로벌 허브 클러스터 지원	관리 허브 클러스터 지원
Red Hat Advanced Cluster Management 2.9 이상 2.9.x 릴리스	제공됨	제공됨
Red Hat Advanced Cluster Management 2.8.3 이상 2.8.x 릴리스	제공됨	제공됨
Red Hat Advanced Cluster Management on Arm	제공됨	제공됨
Red Hat Advanced Cluster Management on IBM Z	제공됨	제공됨
Red Hat Advanced Cluster Management on IBM Power Systems	제공됨	제공됨

- 다중 클러스터 글로벌 허브는 다음 미들웨어를 지원합니다.
  - Kafka 3.4 이상 3.4.x 릴리스
  - PostgreSQL 버전 13 이상 13.x 릴리스

#### 1.2.4. 추가 리소스

- [연결된 환경에서 다중 클러스터 글로벌 허브 설치](#)
- [연결이 끊긴 환경에서 다중 클러스터 글로벌 허브 설치](#)

### 1.3. 연결된 환경에 다중 클러스터 글로벌 허브 설치

다중 클러스터 글로벌 허브는 Operator를 구성하는 구성 요소의 설치, 업그레이드 및 제거를 관리하는 Operator Lifecycle Manager를 통해 설치됩니다.

**필수 액세스:** 클러스터 관리자

#### 1.3.1. 사전 요구 사항

- OpenShift Container Platform Dedicated 환경의 경우 환경에 액세스하려면 **cluster-admin** 권한이 있어야 합니다. 기본적으로 **dedicated-admin** 역할에는 OpenShift Container Platform Dedicated 환경에서 네임스페이스를 생성하는 데 필요한 권한이 없습니다.
- Kubernetes용 Red Hat Advanced Cluster Management를 설치하고 구성해야 합니다. 자세한 내용은 [설치 및 업그레이드](#) 를 참조하십시오.
- Red Hat Advanced Cluster Management 네트워크를 구성해야 합니다. 관리형 허브 클러스터는 Red Hat Advanced Cluster Management에서 다중 클러스터 글로벌 허브의 관리형 클러스터이기도 합니다. 자세한 내용은 [Hub 클러스터 네트워크 구성](#) 을 참조하십시오.

### 1.3.1.1. 콘솔을 사용하여 다중 클러스터 글로벌 허브 설치

OpenShift Container Platform 콘솔을 사용하여 연결된 환경에 다중 클러스터 글로벌 허브 Operator를 설치하려면 다음 단계를 완료하십시오.

1. **cluster-admin** 역할의 사용자로 OpenShift Container Platform 콘솔에 로그인합니다.
2. 탐색 메뉴에서 **Operator > OperatorHub** 아이콘을 선택합니다.
3. **Multicluster** 글로벌 허브 Operator 를 찾아 선택합니다.
4. 설치를 클릭하여 설치를 시작합니다.
5. 설치가 완료되면 *Installed Operators* 페이지에서 상태를 확인합니다.
6. **Multicluster** 글로벌 허브 Operator 를 클릭하여 Operator 페이지로 이동합니다.
7. **Multicluster** 글로벌 허브 탭을 클릭하여 **Multicluster** 글로벌 허브 인스턴스를 확인합니다.
8. **Create Multicluster Global Hub** 를 클릭하여 **Multicluster** 글로벌 허브 인스턴스를 생성합니다.
9. 필요한 정보를 입력하고 **생성** 을 클릭하여 **다중 클러스터 글로벌 허브** 인스턴스를 생성합니다.

#### 참고:

- 다중 클러스터 글로벌 허브는 x86 플랫폼에서만 사용할 수 있습니다.
- 다중 클러스터 글로벌 허브를 설치한 후 Red Hat Advanced Cluster Management에서 정책 및 애플리케이션이 비활성화됩니다.

### 1.3.2. 추가 리소스

- Operator 카탈로그 미러링에 대한 자세한 내용은 Operator 카탈로그 [미러링](#)을 참조하십시오.
- 프라이빗 레지스트리에서 이미지에 액세스하는 방법에 대한 자세한 내용은 프라이빗 레지스트리에서 [Operator의 이미지](#) 액세스를 참조하십시오.
- 카탈로그 소스 추가에 대한 자세한 내용은 [클러스터에 카탈로그 소스 추가](#)를 참조하십시오.
- 연결이 끊긴 환경에서 Red Hat Advanced Cluster Management를 설치하는 방법에 대한 자세한 내용은 [연결이 끊긴 네트워크 환경에 설치](#)를 참조하십시오.
- 이미지 미러링에 대한 자세한 내용은 [연결이 끊긴 설치의 이미지 미러링](#)을 참조하십시오.
- OLM과의 Operator SDK 통합에 대한 자세한 내용은 [Operator SDK와 Operator Lifecycle Manager 통합](#)을 참조하십시오.

## 1.4. 연결이 끊긴 환경에서 다중 클러스터 글로벌 허브 설치

클러스터가 제한된 네트워크에 있는 경우 연결이 끊긴 환경에 다중 클러스터 글로벌 허브 Operator를 배포할 수 있습니다.

**필수 액세스:** 클러스터 관리자

### 1.4.1. 사전 요구 사항

연결이 끊긴 환경에 다중 클러스터 글로벌 허브를 설치하기 전에 다음 요구 사항을 충족해야 합니다.

- 이미지 레지스트리와 bastion 호스트는 인터넷과 미러 레지스트리에 대한 액세스 권한이 있어야 합니다.
- 클러스터에 Operator Lifecycle Manager를 설치합니다. [OLM\(Operator Lifecycle Manager\)](#) 을 참조하십시오.
- Red Hat Advanced Cluster Management for Kubernetes를 설치합니다.
- 다음 명령줄 인터페이스를 설치합니다.
  - OpenShift Container Platform 명령줄입니다. [OpenShift Container Platform CLI 시작하기](#)를 참조하십시오.
  - **opm** 명령줄입니다. [opm CLI 설치](#)를 참조하십시오.
  - **oc-mirror** 플러그인 **oc-plugin**을 사용하여 연결이 끊긴 설치의 이미지 미러링 을 참조하십시오.

## 1.4.2. 미러 레지스트리 구성

연결이 끊긴 환경에서 다중 클러스터 글로벌 허브를 설치하려면 로컬 미러 이미지 레지스트리를 사용해야 합니다. 이 시점에서 OpenShift Container Platform 클러스터 설치 중에 미러 레지스트리를 설정했다고 가정합니다.

다중 클러스터 글로벌 허브의 미러 레지스트리를 프로비저닝하려면 다음 절차를 완료합니다.

### 1.4.2.1. oc-mirror 플러그인을 사용하여 미러 카탈로그에서 Operator 패키지 생성

Red Hat은 [registry.redhat.io/redhat/redhat-operator-index](https://registry.redhat.io/redhat/redhat-operator-index) 인덱스 이미지에서 제공하는 Red Hat Operator 카탈로그에 다중 클러스터 글로벌 허브 및 AMQ Streams Operator를 제공합니다. 이 카탈로그 인덱스 이미지의 미러를 준비할 때 Red Hat에서 제공하는 전체 카탈로그를 미러링하거나 사용하려는 Operator 패키지만 포함된 하위 집합을 미러링할 수 있습니다.

전체 미러 카탈로그를 생성하는 경우 다중 클러스터 글로벌 허브 및 AMQ Streams를 설치하는 데 필요한 모든 패키지가 포함되어 있으므로 특별한 고려 사항이 필요하지 않습니다. 그러나 특정 패키지를 식별하는 부분 또는 필터링된 미러링된 카탈로그를 생성하는 경우 목록에 **multicluster-global-hub-operator-rh** 및 **amq-streams** 패키지 이름을 포함해야 합니다.

**multicluster-global-hub-operator-rh** 및 **amq-streams** 패키지의 로컬 미러 레지스트리를 생성하려면 다음 단계를 완료합니다.

1. **ImageSetConfiguration** YAML 파일을 생성하여 Operator 이미지를 구성하고 추가합니다. YAML 파일은 다음 콘텐츠와 유사할 수 있습니다. 현재 버전은 **4.x** 를 대체합니다.

```
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
storageConfig:
  registry:
    imageURL: myregistry.example.com:5000/mirror/oc-mirror-metadata
mirror:
  platform:
    channels:
      - name: stable-4.x
      type: ocp
```

```

operators:
- catalog: registry.redhat.io/redhat/redhat-operator-index:v4.13
packages:
- name: multicluster-global-hub-operator-rh
- name: amq-streams
additionalImages: []
helm: {}

```

- 다음 명령을 사용하여 대상 미러 레지스트리에 직접 설정된 이미지를 미러링합니다.

```
oc mirror --config=./imageset-config.yaml docker://myregistry.example.com:5000
```

- 완전히 연결이 끊긴 환경에서 이미지 세트를 미러링합니다. 자세한 내용은 [연결이 끊긴 설치의 이미지 미러링을 참조하십시오](#).

#### 1.4.2.2. 연결이 끊긴 클러스터에 레지스트리 및 카탈로그 추가

연결이 끊긴 클러스터에서 미러 레지스트리 및 카탈로그를 사용할 수 있도록 하려면 다음을 수행합니다. 다음 단계를 완료합니다.

- Operator Hub의 기본 카탈로그 소스를 비활성화합니다. 다음 명령을 실행하여 **OperatorHub** 리소스를 업데이트합니다.

```

oc patch OperatorHub cluster --type json \
-p [{"op": "add", "path": "/spec/disableAllDefaultSources", "value": true}]

```

- 절차를 완료하고 Operator 카탈로그 미러링을 통해 **Operator 카탈로그**를 미러링합니다.
- 미러링된 카탈로그의 **CatalogSource** 리소스를 **openshift-marketplace** 네임스페이스에 추가합니다. **CatalogSource** YAML 파일은 다음 예와 유사할 수 있습니다.

```

apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: my-mirror-catalog-source
  namespace: openshift-marketplace
spec:
  image: myregistry.example.com:5000/mirror/my-operator-index:v4.13
  sourceType: grpc
  secrets:
  - <global-hub-secret>

```

- 참고: **metadata.name** 필드의 값을 기록해 둡니다.

- 업데이트된 파일을 저장합니다.
- 사용 가능한 **PackageManifest** 리소스를 쿼리하여 연결이 끊긴 클러스터에서 필요한 패키지를 사용할 수 있는지 확인합니다. 다음 명령을 사용하여 다음 명령을 실행합니다.

```
oc -n openshift-marketplace get packagemanifests
```

표시되는 목록에는 **multicluster-global-hub-operator-rh** 및 **amq-streams** 패키지가 미러 카탈로그의 카탈로그 소스에서 제공됨을 나타내는 항목이 포함되어야 합니다.

### 1.4.3. 이미지 레지스트리 구성

클러스터가 인터넷 호스팅 레지스트리가 아닌 로컬 미러 레지스트리에서 다중 클러스터 글로벌 허브 Operator의 컨테이너 이미지를 가져오려면 미러 레지스트리로 이미지 참조를 리디렉션하도록 연결이 끊긴 클러스터에 **ImageContentSourcePolicy** 리소스를 구성해야 합니다. **ImageContentSourcePolicy** 는 이미지 **다이제스트** 가 있는 이미지 미러만 지원합니다.

**oc adm catalog mirror** 명령을 사용하여 카탈로그를 미러링한 경우 필요한 이미지 콘텐츠 소스 정책 구성은 해당 명령으로 생성된 **manifests-** 디렉터리 내부의 **imageContentSourcePolicy.yaml** 파일에 있습니다.

**oc-mirror** 플러그인을 사용하여 카탈로그를 미러링한 경우 **imageContentSourcePolicy.yaml** 파일은 **oc-mirror-workspace/results-** 디렉터리에서 **oc-mirror -workspace/results-** 디렉터리에 있습니다.

두 경우 모두 **oc apply** 또는 **oc replace** 명령을 사용하여 연결이 끊긴 명령에 정책을 적용할 수 있습니다 (예: **oc replace -f ./<path>/imageContentSourcePolicy.yaml**).

필요한 이미지 콘텐츠 소스 정책 설명은 미러 레지스트리를 생성한 방법에 따라 다를 수 있지만 다음 예와 유사합니다.

```
apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  labels:
    operators.openshift.org/catalog: "true"
  name: global-hub-operator-icsp
spec:
  repositoryDigestMirrors:
  - mirrors:
    - myregistry.example.com:5000/multicluster-globalhub
    source: registry.redhat.io/multicluster-globalhub
  - mirrors:
    - myregistry.example.com:5000/openshift4
    source: registry.redhat.io/openshift4
  - mirrors:
    - myregistry.example.com:5000/redhat
    source: registry.redhat.io/redhat
```

**ManagedClusterImageRegistry** 를 사용하여 다양한 관리 허브에 대해 다양한 이미지 레지스트리를 구성할 수 있습니다. **ManagedClusterImageRegistry** API를 사용하여 에이전트 이미지를 교체하려면 **ManagedClusterImageRegistry** 가 있는 클러스터 가져오기 를 참조하십시오.

이전 단계를 완료하면 레이블과 주석이 선택한 **ManagedCluster** 에 추가됩니다. 즉 클러스터의 에이전트 이미지가 미러 이미지로 교체됩니다.

- 레이블: **multicluster-global-hub.io/image-registry=**  
**<namespace.managedclusterimageregistry-name>**
- 주석: **multicluster-global-hub.io/image-registries: <image-registry-info>**

#### 1.4.3.1. 이미지 풀 시크릿 구성

구독한 Operator에서 참조하는 Operator 또는 Operand 이미지에 프라이빗 레지스트리에 액세스해야 하는 경우 클러스터의 모든 네임스페이스 또는 개별 대상 테넌트 네임스페이스에 대한 액세스 권한을 제공할 수 있습니다.

### 1.4.3.1.1. OpenShift Container Platform 클러스터에서 다중 클러스터 글로벌 허브 이미지 풀 시크릿 구성

기존 OpenShift Container Platform 클러스터에서 이미지 풀 시크릿을 구성할 수 있습니다.

**참고:** 기존 클러스터에 이미지 풀 시크릿을 적용하면 모든 노드가 롤링 재시작됩니다.

풀 시크릿을 구성하려면 다음 단계를 완료합니다.

1. 풀 시크릿에서 사용자 이름을 내보냅니다.

```
export USER=<the-registry-user>
```

2. 풀 시크릿에서 암호를 내보냅니다.

```
export PASSWORD=<the-registry-password>
```

3. 풀 시크릿을 복사합니다.

```
oc get secret/pull-secret -n openshift-config --template='{{index .data ".dockerconfigjson" | base64decode}}' > pull_secret.yaml
```

4. 풀 시크릿을 사용하여 로그인합니다.

```
oc registry login --registry=${REGISTRY} --auth-basic="$USER:$PASSWORD" --to=pull_secret.yaml
```

5. 다중 클러스터 글로벌 허브 이미지 풀 시크릿을 지정합니다.

```
oc set data secret/pull-secret -n openshift-config --from-file=.dockerconfigjson=pull_secret.yaml
```

6. 이전 풀 시크릿을 제거합니다.

```
rm pull_secret.yaml
```

### 1.4.3.1.2. 개별 네임스페이스에 대한 다중 클러스터 글로벌 허브 이미지 풀 시크릿 구성

다음 단계를 완료하여 개별 네임스페이스에 이미지 가져오기 보안을 구성할 수 있습니다.

1. 다음 명령을 실행하여 테넌트 네임스페이스에 보안을 생성합니다.

```
oc create secret generic <secret_name> -n <tenant_namespace> \
--from-file=.dockerconfigjson=<path/to/registry/credentials> \
--type=kubernetes.io/dockerconfigjson
```

2. Operator 또는 피연산자의 서비스 계정에 보안을 연결합니다.

```
oc secrets link <operator_sa> -n <tenant_namespace> <secret_name> --for=pull
```

### 1.4.3.2. Global Hub Operator 설치



Red Hat OpenShift Container Platform 웹 콘솔을 사용하여 OperatorHub에서 Operator를 설치하고 구동할 수 있습니다. 절차를 보려면 클러스터에 [Operator 추가](#) 를 참조하십시오. Operator를 추가한 후 다음 명령을 실행하여 multicluster 글로벌 허브 Operator의 상태를 확인할 수 있습니다.

```
oc get pods -n multicluster-global-hub
NAME                                READY STATUS  RESTARTS  AGE
multicluster-global-hub-operator-687584cb7c-fnftj  1/1    Running  0         2m12s
```

#### 1.4.4. 추가 리소스

- 미리 레지스트리 생성에 대한 자세한 내용은 미리 레지스트리 [만들기](#)를 참조하십시오.
- 이미지 미러링에 대한 자세한 내용은 [연결이 끊긴 설치의 이미지 미러링](#)을 참조하십시오.
- Operator 카탈로그 미러링에 대한 자세한 내용은 Operator 카탈로그 [미러링](#)을 참조하십시오.

### 1.5. 기존 구성 요소 통합

다중 클러스터 글로벌 허브에는 정책 규정 준수 보기를 제공하기 위해 Grafana와 함께 미들웨어 구성 요소, Kafka 및 PostgreSQL이 필요합니다. 다중 클러스터 글로벌 허브는 Kafka, PostgreSQL, Grafana 버전을 제공합니다. 기존 Kafka, PostgreSQL 및 Grafana를 통합할 수도 있습니다.

- [기존 Kafka 버전 통합](#)
- [기존 버전의 PostgreSQL 통합](#)
- [기존 Grafana 버전 통합](#)

#### 1.5.1. 기존 Kafka 버전 통합

고유한 Kafka 인스턴스가 있는 경우 다중 클러스터 글로벌 허브의 전송으로 사용할 수 있습니다. Kafka 3.3은 테스트된 버전입니다. Kafka 인스턴스를 통합하려면 다음 단계를 완료합니다.

1. Kafka 인스턴스에 대한 영구 볼륨이 없는 경우 이를 생성해야 합니다.
2. **multicluster-global-hub** 네임스페이스에서 **multicluster-global-hub-transport** 라는 시크릿을 생성합니다.
  - a. 다음 필수 필드에서 정보를 추출합니다.
    - **bootstrap.servers**: Kafka 부트스트랩 서버를 지정합니다.
    - **ca.crt: KafkaUser** 사용자 정의 리소스를 사용하여 인증 인증 정보를 구성하는 경우 필요합니다. 시크릿에서 **ca.crt** 인증서를 추출하는 데 필요한 단계는 STRIMZI 설명서의 [사용자 인증 주제](#)를 참조하십시오.
    - **client.crt**: 필수는 시크릿에서 **user.crt** 인증서를 추출하는 단계는 STRIMZI 설명서의 [사용자 인증 주제](#)를 참조하십시오.
    - **client.key**: 필수는 시크릿에서 **user.key** 를 추출하는 단계는 STRIMZI 설명서의 [사용자 인증 주제](#)를 참조하십시오.
3. 다음 명령을 실행하여 시크릿을 생성하고 필요한 경우 값을 추출된 값으로 교체합니다.

```
oc create secret generic multicluster-global-hub-transport -n multicluster-global-hub \
```

```
--from-literal=bootstrap_server=<kafka-bootstrap-server-address> \
--from-file=ca.crt=<CA-cert-for-kafka-server> \
--from-file=client.crt=<Client-cert-for-kafka-server> \
--from-file=client.key=<Client-key-for-kafka-server>
```

4. Kafka 인스턴스에 자동 주제 생성이 구성된 경우 이 단계를 건너뛰니다. 구성되지 않은 경우 **사양,상태 및 이벤트** 주제를 수동으로 생성합니다.
5. Kafka에 액세스하는 글로벌 허브 사용자에게 주제에서 데이터를 읽고 주제에 데이터를 쓸 수 있는 권한이 있는지 확인합니다.

### 1.5.2. 기존 버전의 PostgreSQL 통합

PostgreSQL 관계형 데이터베이스가 있는 경우 이를 다중 클러스터 글로벌 허브의 스토리지로 사용할 수 있습니다. PostgreSQL 13은 테스트된 버전입니다.

필요한 최소 스토리지 크기는 20GB입니다. 이 용량은 18개월 동안 250개의 관리 클러스터와 관리 허브당 50개의 정책이 포함된 3개의 관리 허브를 저장할 수 있습니다. **multicluster-global-hub** 네임스페이스에 **multicluster-global-hub-storage** 라는 시크릿을 생성해야 합니다. 보안에는 다음 필드가 포함되어야 합니다.

- **database\_uri**: 데이터베이스를 생성하고 데이터를 삽입하는 데 사용됩니다. 값은 **postgres://<user>:<password>@<host>:<port>/<database>?sslmode=<mode>** 와 유사해야 합니다.
- **database\_uri\_with\_readonlyuser**: multicluster 글로벌 허브에서 사용하는 Grafana 인스턴스에서 데이터를 쿼리하는 데 사용됩니다. 선택적 값입니다. 값은 **postgres://<user>:<password>@<host>:<port>/<database>?sslmode=<mode>** 와 유사해야 합니다.
- **sslmode** 를 기반으로 하는 **ca.crt** 는 선택적 값입니다.
  1. 클러스터에 필요한 최소 스토리지 크기가 20GB인지 확인합니다. 이 용량은 18개월 동안 250개의 관리 클러스터와 관리 허브당 50개의 정책이 포함된 3개의 관리 허브를 저장할 수 있습니다.
  2. 다음 명령을 실행하여 시크릿을 생성합니다.

```
oc create secret generic multicluster-global-hub-storage -n multicluster-global-hub \
--from-literal=database_uri=<postgresql-uri> \
--from-literal=database_uri_with_readonlyuser=<postgresql-uri-with-readonlyuser> \
--from-file=ca.crt=<CA-for-postgres-server>
```

호스트는 multicluster 글로벌 허브 클러스터에서 액세스할 수 있어야 합니다. PostgreSQL 데이터베이스가 Kubernetes 클러스터에 있는 경우 **nodePort** 또는 **LoadBalancer** 와 함께 서비스 유형을 사용하여 데이터베이스를 노출하는 것이 좋습니다. 자세한 내용은 [문제 해결을 위해 프로비저닝된 postgres 데이터베이스 액세스](#)를 참조하십시오.

### 1.5.3. 기존 Grafana 버전 통합

기존 Grafana 인스턴스를 사용하면 자체 Grafana에 의존하는 경우 다양한 클러스터에서 Prometheus와 같은 여러 소스의 지표를 가져오고 메트릭을 직접 집계하는 경우 다중 클러스터 글로벌 허브에서 작동할 수 있습니다. 다중 클러스터 글로벌 허브 데이터를 자체 Grafana로 가져오려면 데이터 소스를 구성하고 대시보드를 가져와야 합니다.

1. 다음 명령을 실행하여 다중 클러스터 글로벌 허브 Grafana 데이터 소스 시크릿에서 PostgreSQL 연결 정보를 수집합니다.

```
oc get secret multicluster-global-hub-grafana-datasources -n multicluster-global-hub -ojsonpath='{.data.datasources\.yaml}' | base64 -d
```

출력은 다음 예와 유사합니다.

```
apiVersion: 1
datasources:
- access: proxy
  isDefault: true
  name: Global-Hub-DataSource
  type: postgres
  url: postgres-primary.multicluster-global-hub.svc:5432
  database: hoh
  user: guest
jsonData:
  sslmode: verify-ca
  tlsAuth: true
  tlsAuthWithCACert: true
  tlsConfigurationMethod: file-content
  tlsSkipVerify: true
  queryTimeout: 300s
  timeInterval: 30s
secureJsonData:
  password: xxxxx
  tlsCACert: xxxxx
```

2. PostgreSQL과 같은 소스를 추가하여 자체 Grafana 인스턴스에서 데이터 소스를 구성하고 이전에 추출한 정보로 필수 필드를 완료합니다.

다음 필수 필드를 참조하십시오.

- 이름
- 호스트
- 데이터베이스
- 사용자
- 암호
- TLS/SSL 모드
- TLS/SSL 방법
- CA 인증서

3. Grafana가 다중 클러스터 글로벌 허브 클러스터에 없는 경우 외부에서 PostgreSQL에 액세스할 수 있도록 **LoadBalancer** 를 사용하여 PostgreSQL을 노출해야 합니다. **PostgresCluster** 피연산자에 다음 값을 추가할 수 있습니다.

```
service:
  type: LoadBalancer
```

해당 콘텐츠를 추가한 후 **postgres-ha** 서비스에서 **EXTERNAL-IP** 를 가져올 수 있습니다. 다음 예제를 참조하십시오.

```
oc get svc postgres-ha -n multicluster-global-hub
NAME      TYPE          CLUSTER-IP    EXTERNAL-IP          PORT(S)          AGE
postgres-ha LoadBalancer  172.30.227.58  xxxx.us-east-1.elb.amazonaws.com  5432:31442/TCP  128m
```

해당 명령을 실행한 후 PostgreSQL 연결 호스트로 **xxxx.us-east-1.elb.amazonaws.com:5432** 를 사용할 수 있습니다.

#### 4. 기존 대시보드를 가져옵니다.

- 공식 Grafana 문서의 [대시보드 내보내기 및 가져오기](#) 단계에 따라 기존 Grafana 인스턴스에서 대시보드를 내보냅니다.
- 공식 Grafana 설명서의 [내보내기 및 가져오기 대시보드](#) 단계에 따라 대시보드를 다중 클러스터 글로벌 허브 Grafana 인스턴스로 가져옵니다.

### 1.5.4. 추가 리소스

시크릿에서 **ca.crt** 인증서를 추출하는 방법에 대한 자세한 내용은 STRIMZI 설명서의 [사용자 인증](#) 을 참조하십시오.

시크릿에서 **user.crt** 인증서를 추출하는 단계는 STRIMZI 설명서의 [사용자 인증](#) 을 참조하십시오.

## 1.6. 기본 모드에서 관리 허브 클러스터 가져오기

기존 허브 클러스터를 관리 허브 클러스터로 가져오려면 다음 단계를 완료합니다.

- multiclusterhub** 사용자 정의 리소스에서 **disableHubSelfManagement** 설정을 **true** 로 설정하여 기존 Red Hat Advanced Cluster Management for Kubernetes Hub 클러스터에서 클러스터 자체 관리를 비활성화합니다. 이 설정은 hub 클러스터를 관리 클러스터 자동 가져오기를 비활성화합니다.
- 클러스터 가져오기 [도입 단계를 완료하여 관리 허브 클러스터를 가져옵니다.](#)
- 관리 허브 클러스터를 가져온 후 multicluster 글로벌 허브 에이전트 상태를 확인하여 다음 명령을 실행하여 에이전트가 관리 허브 클러스터에서 실행 중인지 확인합니다.

```
oc get managedclusteraddon multicluster-global-hub-controller -n
<managed_hub_cluster_name>
```

## 1.7. GRAFANA 데이터 액세스

Grafana 데이터는 경로를 통해 노출됩니다. 다음 명령을 실행하여 로그인 URL을 표시합니다.

```
oc get route multicluster-global-hub-grafana -n <the-namespace-of-multicluster-global-hub-instance>
```

이 URL의 인증 방법은 Red Hat OpenShift Container Platform 콘솔에 대한 인증과 동일합니다.

### 1.7.1. Grafana 대시보드를 사용하여 정책 상태 보기

글로벌 허브 Grafana 데이터에 액세스한 후 관리하는 허브 클러스터 환경을 통해 구성된 정책을 모니터링할 수 있습니다.

다중 클러스터 글로벌 허브 대시보드에서 선택한 기간 동안 시스템 정책의 규정 준수 상태를 확인할 수 있습니다. 정책 규정 준수 상태는 매일 업데이트되므로 대시보드에 다음 날까지 현재 날짜의 상태가 표시되지 않습니다.

다중 클러스터 글로벌 허브 대시보드를 탐색하려면 정책 또는 클러스터 별로 정책 데이터를 그룹화하여 정책 데이터를 관찰하고 필터링할 수 있습니다.

정책 그룹화를 사용하여 정책 데이터를 검사하려면 **Global Hub - 정책 그룹 호환 개요** 라는 대시보드에서 시작합니다.

이 대시보드를 사용하면 표준, 범주 및 제어에 따라 정책 데이터를 필터링할 수 있습니다. 그래프에서 특정 시점을 선택하면 **Global Hub - Offending Policies** 대시보드로 이동합니다. 글로벌 허브 - 정책 오프로드 대시보드에는 해당 시점에 호환되지 않거나 알 수 없는 정책이 나열됩니다. 대상 정책을 선택한 후 관련 이벤트를 보고 글로벌 허브 - **What's Changed / Policies** 대시보드에 액세스하여 변경된 사항을 확인할 수 있습니다.

마찬가지로 클러스터 그룹화로 정책 데이터를 검사하려면 **Global Hub - Cluster Group Compliancy Overview** 대시보드를 사용하여 시작합니다. 탐색 흐름은 정책 그룹화 흐름과 동일하지만 관리 클러스터 레이블 및 값 과 같이 클러스터와 관련된 필터를 선택합니다. 글로벌 허브 에 도달한 후 모든 클러스터에 대한 정책 이벤트를 보는 대신 개별 클러스터와 관련된 정책 이벤트를 볼 수 있습니다.

## 1.8. GRAFANA 경고 (기술 프리뷰)

**multicluster-global-hub-default-alerting** 구성 맵에 저장된 Grafana 경고 3개를 구성할 수 있습니다. 이 경고는 의심스러운 정책, 의심스러운 클러스터 준수 상태 변경 및 실패한 cron 작업을 알려줍니다.

경고에 대한 다음 설명을 참조하십시오.

- 의심스러운 정책 변경: 이 경고 규칙은 의심스러운 정책 변경을 감시 합니다. 다음 이벤트가 1시간에 5회 이상 발생하면 알림이 생성됩니다.
  - 정책이 활성화되거나 비활성화되었습니다.
  - 정책이 업데이트되었습니다.
- 의심스러운 클러스터 규정 준수 상태 변경: 이 경고 규칙은 클러스터의 클러스터 규정 준수 상태 및 정책 이벤트를 감시합니다. 이 경고에는 다음 두 가지 규칙이 있습니다.

- 클러스터 규정 준수 상태가 자주 변경됩니다. 클러스터 규정 준수 상태가 규정 준수 에서 1시간 동안 3회 이상 변경되는 경우 알림이 생성됩니다.
- 클러스터에 너무 많은 정책 이벤트: 클러스터의 정책의 경우 5분 내에 이벤트가 20개 이상 있는 경우 알림이 생성됩니다. 이 경고가 항상 실행되는 경우 `event.local_policies` 테이블의 데이터가 너무 빨리 증가합니다.
- **cron 작업 실패:** 이 경고는 실패한 이벤트에 대한 cron 작업 구성에 설명된 cron 작업을 감시합니다. 이 경고에는 다음 두 가지 규칙이 있습니다.
  - 로컬 규정 준수 작업 실패: 이 경고 규칙이 알림을 생성하면 로컬 규정 준수 상태 동기화 작업이 실패했음을 의미합니다. `history.local_compliance` 테이블의 데이터가 손실될 수 있습니다. 필요한 경우 작업을 수동으로 실행합니다.
  - 데이터 보존 작업에 실패했습니다. 이 경고 규칙이 알림 생성을 시작하면 데이터 보존 작업이 실패했음을 의미합니다. 수동으로 실행할 수 있습니다.

### 1.8.1. 기본 Grafana 경고 규칙 삭제

기본 Grafana 경고 규칙이 유용한 정보를 제공하지 않으면 `multicluster-global-hub-custom-alerting` 구성 맵에 `deleteRules` 섹션을 포함하여 Grafana 경고 규칙을 삭제할 수 있습니다. `multicluster-global-hub-custom-alerting` 구성 맵에 대한 자세한 내용은 Grafana 경고 리소스 사용자 지정을 참조하십시오.

기본 경고를 모두 삭제하려면 `deleteRules` 구성 섹션은 다음 예와 유사해야 합니다.

```
deleteRules:
- orgId: 1
  uid: globalhub_suspicious_policy_change
- orgId: 1
  uid: globalhub_cluster_compliance_status_change_frequently
- orgId: 1
  uid: globalhub_high_number_of_policy_events
- orgId: 1
  uid: globalhub_data_retention_job
- orgId: 1
  uid: globalhub_local_compliance_job
```

### 1.8.2. Grafana 경고 사용자 정의

다중 클러스터 글로벌 허브는 사용자 정의 **Grafana** 경고 생성을 지원합니다. **Grafana** 경고를 사용자 지정하려면 다음 단계를 완료합니다.

### 1.8.2.1. grafana.ini 파일 사용자 정의

**grafana.ini** 파일을 사용자 지정하려면 다중 클러스터 글로벌 허브 **Operator**를 설치한 네임스페이스에 **multicluster-global-hub-custom-grafana-config** 라는 시크릿을 생성합니다. 다음 예와 같이 **secret** 데이터 키는 **grafana.ini** 입니다. 필요한 정보를 자체 인증 정보로 교체합니다.

```
apiVersion: v1
kind: Secret
metadata:
  name: multicluster-global-hub-custom-grafana-config
  namespace: multicluster-global-hub
type: Opaque
stringData:
  grafana.ini: |
    [smtp]
    enabled = true
    host = smtp.google.com:465
    user = <example@google.com>
    password = <password>
    ;cert_file =
    ;key_file =
    skip_verify = true
    from_address = <example@google.com>
    from_name = Grafana
    ;ehlo_identity = dashboard.example.com ①
```

<1> SMTP 대화 상자의 EHLO ID는 기본적으로 **instance\_name** 입니다.

참고: 이미 **multicluster-global-hub-default-grafana-config** 시크릿이 포함된 섹션을 구성할 수 없습니다.

### 1.8.2.2. Grafana 경고 리소스 사용자 정의

다중 클러스터 글로벌 허브는 **Grafana** 문서의 **파일 프로비저닝을 사용하여 경고 리소스 생성 및 관리**에 설명된 **경고 리소스** 사용자 지정을 지원합니다.

경고 리소스를 사용자 지정하려면 **multicluster-global-hub -hub** 네임스페이스에서 **multicluster-global-hub-custom-alerting** 이라는 구성 맵을 생성합니다.

구성 맵 데이터 키는 다음 예와 같이 **alert.yaml**입니다.

```

apiVersion: v1
data:
  alerting.yaml: |
    contactPoints:
      - orgId: 1
        name: globalhub_policy
        receivers:
          - uid: globalhub_policy_alert_email
            type: email
            settings:
              addresses: <example@redhat.com>
              singleEmail: false
          - uid: globalhub_policy_alert_slack
            type: slack
            settings:
              url: <Slack-webhook-URL>
              title: |
                {{ template "globalhub.policy.title" . }}
              text: |
                {{ template "globalhub.policy.message" . }}
        policies:
          - orgId: 1
            receiver: globalhub_policy
            group_by: ['grafana_folder', 'alertname']
            matchers:
              - grafana_folder = Policy
            repeat_interval: 1d
        deleteRules:
          - orgId: 1
            uid: [Alert Rule Uid]
        muteTimes:
          - orgId: 1
            name: mti_1
            time_intervals:
              - times:
                  - start_time: '06:00'
                    end_time: '23:59'
                    location: 'UTC'
                  weekdays: ['monday:wednesday', 'saturday', 'sunday']
                  months: ['1:3', 'may:august', 'december']
                  years: ['2020:2022', '2030']
                  days_of_month: ['1:5', '-3:-1']
    kind: ConfigMap
    metadata:
      name: multicluster-global-hub-custom-alerting
      namespace: multicluster-global-hub

```

## 1.9. CRON 작업 구성



다중 클러스터 글로벌 허브의 **cron** 작업 설정을 구성할 수 있습니다.

다중 클러스터 글로벌 허브 피연산자를 설치하면 다중 클러스터 글로벌 허브 관리자가 실행되고 다음 **cron** 작업을 예약할 수 있는 작업 스케줄러가 표시됩니다.

- 로컬 규정 준수 상태 동기화 작업: 이 **cron** 작업은 이전 날 관리자가 수집한 정책 상태 및 이벤트에 따라 매일 자정에 실행됩니다. 이 작업을 실행하면 클러스터의 정책 변경 상태 및 변경 빈도가 요약되어 **history.local\_compliance** 테이블에 **Grafana** 대시보드의 데이터 소스로 저장됩니다.
- 데이터 보존 작업: 멀티 클러스터 글로벌 허브의 일부 데이터 테이블은 시간이 지남에 따라 계속 증가하므로 일반적으로 테이블이 너무 클 때 문제가 발생할 수 있습니다. 다음 두 가지 방법은 너무 큰 테이블의 결과 문제를 최소화하는 데 도움이 됩니다.
  - 더 이상 필요하지 않은 오래된 데이터 삭제
  - 큰 테이블의 파티션으로 쿼리 및 삭제를 더 빠르게 실행 가능

**event.local\_policies** 및 **history.local\_compliance** 와 같은 이벤트 테이블의 경우 매일 크기가 증가하는 **history.local\_compliance**는 큰 테이블을 더 작은 파티션으로 나눕니다. 이 프로세스는 실행될 때마다 다음 달의 파티션 테이블도 생성합니다. **local\_spec.policies** 및 **status.managed\_clusters** 와 같은 정책 및 클러스터 테이블의 경우 테이블에 **deleted\_at** 인덱스가 있어 하드 삭제할 때 성능을 향상시킬 수 있습니다.

다중 클러스터 글로벌 허브 피연산자의 보존 설정을 변경하여 데이터가 유지되는 기간을 변경할 수 있습니다. 권장 최소값은 1 개월이며 기본값은 18 개월입니다. 이 작업의 실행 간격은 1개월 미만이어야 합니다.

나열된 **cron** 작업은 다중 클러스터 글로벌 허브 관리자가 시작될 때마다 실행됩니다. 로컬 규정 준수 상태 동기화 작업은 하루에 한 번 실행되며 결과를 변경하지 않고 하루 이내에 여러 번 실행할 수 있습니다. 데이터 보존 작업은 일주일에 한 번 실행되며 결과를 변경하지 않고 한 달에 여러 번 실행할 수도 있습니다.

이러한 작업의 상태는 **Red Hat OpenShift Container Platform** 클러스터 콘솔에서 볼 수 있는 **multicluster\_global\_hub\_jobs\_status** 라는 메트릭에 저장됩니다. 값 0 은 작업이 성공적으로 실행되었지만 값 1 은 실패했음을 나타냅니다.

실패한 작업이 있는 경우 로그 테이블 (`history.local_compliance_job_log,event.data_retention_job_log`)을 사용하여 문제를 해결할 수 있습니다. 자세한 내용 및 서비스 실행 여부를 결정하는 지침은 [규정 준수 데이터 복원을 참조하십시오](#).

### 1.10. 요약 프로세스 수동 실행

요약 프로세스를 수동으로 실행할 수도 있습니다. 이 기능은 문제를 조사하거나 다음 스케줄링된 루틴보다 빨리 보고서를 필요로 할 때 유용할 수 있습니다.

수동 요약 프로세스는 다음 두 가지 하위 작업으로 구성됩니다.

- `materialized View local_compliance_view_<yyyy_MM_dd>`에서 `history.local_compliance`에 해당 날짜의 클러스터 정책 데이터를 삽입합니다.
- `event.local_policies`를 기반으로 `history.local_compliance`로 날짜의 규정 준수 및 정책 전환 빈도를 업데이트합니다.

요약 프로세스를 수동으로 실행하려면 다음 단계를 완료합니다.

1. 데이터베이스에 연결합니다.

`pgAdmin`, `table Cryostath` 등의 클라이언트를 사용하여 `multicluster` 글로벌 허브 데이터베이스에 연결하여 다음 몇 단계에서 `SQL` 문을 실행할 수 있습니다. 다음 명령을 실행하여 클러스터의 데이터베이스에 직접 연결할 수 있습니다.

```
oc exec -it multicluster-global-hub-postgres-0 -n multicluster-global-hub -- psql -d hoh
```

2. `2023-07-06`과 같이 실행해야 하는 날짜를 확인합니다.

`2023-07-06`의 대시보드에 규정 준수 정보가 없는 경우, `history.local_compliance_job_log`에서 이 날짜의 작업 실패 정보를 찾습니다. 이 경우 `2023-07-07`입니다. `2023-07-06`은 요약 프로세스를 수동으로 실행해야 하는 날짜임을 확인할 수 있습니다.

3. 다음 명령을 실행하여 `history.local_compliance_view_2023_07_06`의 `materialized View`가 존재하는지 확인합니다.

```
select * from history.local_compliance_view_2023_07_06;
```

보기가 있는 경우 다음 명령을 실행하여 보기 레코드를 **history.local\_compliance** 에 로드합니다.

```
-- exec the insert func for that day '2023_07_06'
SELECT history.insert_local_compliance_job('2023_07_06');
```

보기가 없는 경우 해당 날짜 전날의 기록 규정 준수 레코드를 상속합니다. 이 예에서는 **2023\_07\_05** 일 수 있습니다.

```
-- call the func to generate the data of '2023_07_06' by inheriting '2023_07_05'
CALL history.inherit_local_compliance_job('2023_07_05', '2023_07_06');
```

4.

해당 날짜의 규정 준수 및 빈도 정보를 **history.local\_compliance** 로 업데이트합니다.

```
-- call the func to update records start with '2023-07-06', end with '2023-07-07'
SELECT history.update_local_compliance_job('2023_07_06', '2023_07_07');
```

5.

**history.local\_compliance** 에서 생성된 해당 날짜의 레코드를 찾습니다. 다음 명령을 실행하여 **materialized View history.local\_compliance\_view\_2023\_07\_06** 을 안전하게 삭제할 수 있습니다.

```
DROP MATERIALIZED VIEW IF EXISTS history.local_compliance_view_2023_07_06;
```

## 1.11. 다중 클러스터 글로벌 허브 백업 (기술 프리뷰)

**Red Hat Advanced Cluster Management** 백업과 함께 멀티 클러스터 글로벌 허브를 사용하고 복구 솔루션을 위한 복원 기능과 기본 리소스에 액세스합니다. 이러한 기능이 도움이 되는 방법에 대한 자세한 내용은 [백업 및 복원](#) 을 참조하십시오.

다중 클러스터 글로벌 허브는 **acm-hub-pvc-backup** 을 사용하여 백업 **postgres pvc** 도 지원합니다. 다중 클러스터 글로벌 허브가 백업 **postgres pvc** 를 지원할 수 있도록 하려면 현재 버전의 **CryostatlySync** 및 **Red Hat Advanced Cluster Management**가 있어야 합니다. 데이터 백업에 대한 자세한 단계는 [acm-hub-pvc-backup](#) 을 참조하십시오.

### 1.11.1. 다중 클러스터 글로벌 허브 백업 및 복원

다중 클러스터 글로벌 허브 클러스터를 복원해야 하는 경우 **새 허브 클러스터 준비** 를 참조하십시오. 다중 클러스터 글로벌 허브 **Operator**를 설치하지만 **CR**이 자동으로 복원되므로 다중 클러스터 글로벌 허브

**CR(사용자 정의 리소스)을 생성하지 않습니다.**