



# OpenShift Container Platform 4.6

## 로깅

OpenShift Container Platform에서 클러스터 로깅 구성



## OpenShift Container Platform 4.6 로깅

---

OpenShift Container Platform에서 클러스터 로깅 구성

## 법적 공지

Copyright © 2020 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

이 문서는 다양한 OpenShift Container Platform 서비스에 대한 로그를 집계하는 클러스터 로깅 설치, 구성 및 사용에 대한 지침을 제공합니다.

## 차례

<b>1장. 클러스터 로깅 이해</b>	<b>4</b>
1.1. 클러스터 로깅 배포 정보	4
<b>2장. 클러스터 로깅 배포</b>	<b>7</b>
2.1. 웹 콘솔을 사용하여 클러스터 로깅 설치	7
2.2. CLI를 사용하여 클러스터 로깅 설치	11
2.3. 추가 리소스	18
<b>3장. 클러스터 로깅 배포 구성</b>	<b>19</b>
3.1. 클러스터 로깅 사용자 정의 리소스 정보	19
3.2. 로깅 수집기 구성	20
3.3. 로그 저장소 구성	26
3.4. 로그 시각화 프로그램 구성	40
3.5. 클러스터 로깅 스토리지 구성	42
3.6. 클러스터 로깅 구성 요소에 대한 CPU 및 메모리 제한 구성	43
3.7. 공차를 사용하여 클러스터 로깅 포드 배치 제어	44
3.8. 노드 선택기로 클러스터 로깅 리소스 이동	49
3.9. SYSTEMD-JOURNALD 및 FLUENTD 구성	53
3.10. 로그 큐레이터 구성	55
3.11. 유지보수 및 지원	58
<b>4장. 클러스터 로그 보기</b>	<b>61</b>
4.1. 클러스터 로그 보기	61
4.2. KIBANA를 사용하여 클러스터 로그 보기	62
<b>5장. 타사 시스템에 로그 전달</b>	<b>64</b>
5.1. 타사 시스템으로 로그 전달 정보	64
<b>6장. KUBERNETES 이벤트 수집 및 저장</b>	<b>83</b>
6.1. 이벤트 라우터 배포 및 구성	83
<b>7장. 클러스터 로깅 배포</b>	<b>87</b>
7.1. 클러스터 로깅 배포	87
7.2. 로그 전달 사용자 정의 리소스 업데이트	91
<b>8장. 클러스터 대시보드 보기</b>	<b>96</b>
8.1. ELASTISEARCH 및 OPENSIFT 로깅 대시보드에 액세스	96
8.2. OPENSIFT 로깅 대시보드 정보	96
8.3. 로깅/ELASTICSEARCH 노드 대시보드의 차트	98
<b>9장. 클러스터 로깅 문제 해결</b>	<b>104</b>
9.1. 클러스터 로깅 상태 보기	104
9.2. 로그 저장소의 상태 보기	109
9.3. 클러스터 로깅 경고 이해	116
9.4. 로그 시각화 프로그램 문제 해결	118
9.5. 로그 CURATOR 문제 해결	119
<b>10장. 클러스터 로깅 삭제</b>	<b>121</b>
10.1. OPENSIFT CONTAINER PLATFORM에서 클러스터 로깅 설치 삭제	121
<b>11장. 내보낸 필드</b>	<b>122</b>
11.1. 기본 내보낸 필드	122
11.2. SYSTEMD 내보낸 필드	137
11.3. KUBERNETES 내보낸 필드	139

11.4. 컨테이너 내보낸 필드	140
11.5. OVIRT 내보낸 필드	141
11.6. AUSHAPE 내보낸 필드	142
11.7. TLOG 내보낸 필드	143



## 1장. 클러스터 로깅 이해

클러스터 관리자는 클러스터 로깅을 배포하여 OpenShift Container Platform 클러스터의 모든 로그(예: 노드 시스템 감사 로그, 애플리케이션 컨테이너 로그 및 인프라 로그)를 집계할 수 있습니다. 클러스터 로깅은 클러스터 전체에서 이러한 로그를 집계하여 기본 로그 저장소에 저장합니다. [콘솔 또는 OpenShift Container Platform 웹 콘솔에서 로그를 보거나 Kibana를 사용하여 로그 데이터를 시각화](#)할 수 있습니다.

### 1.1. 클러스터 로깅 배포 정보

OpenShift Container Platform 클러스터 관리자는 OpenShift Container Platform 웹 콘솔 또는 CLI를 사용하여 클러스터 로깅을 배포하여 Elasticsearch Operator 및 Cluster Logging Operator를 설치할 수 있습니다. Operator가 설치되면 클러스터 로깅 사용자 정의 리소스(CR)를 생성하여 클러스터 로깅 포트 및 클러스터 로깅을 지원하는 데 필요한 기타 리소스를 예약합니다. Operator는 클러스터 로깅을 배포, 업그레이드 및 유지 관리해야 합니다.

클러스터 로깅 CR은 로그를 수집, 저장 및 시각화하기 위해 로깅 스택의 모든 구성 요소를 포함하는 완전한 클러스터 로깅 배포를 정의합니다. Cluster Logging Operator는 **ClusterLogging** 사용자 정의 리소스를 감시하고 그에 따라 로깅 배포를 조정합니다.

관리자와 애플리케이션 개발자는 보기 권한이 있는 프로젝트의 로그를 볼 수 있습니다.

#### 1.1.1. 클러스터 로깅 구성 요소 정보

클러스터 로깅 구성 요소에는 OpenShift Container Platform 클러스터의 각 노드에 배포된 수집기가 포함되어 모든 노드와 컨테이너 로그를 수집하여 로그 저장소에 작성합니다. 중앙 집중식 웹 UI를 사용하여 집계된 데이터로 풍부한 시각화 및 대시보드를 만들 수 있습니다.

클러스터 로깅의 주요 구성 요소는 다음과 같습니다.

- 수집 - 클러스터에서 로그를 수집하고 형식을 지정 한 후 로그 저장소로 전달하는 구성 요소입니다. 현재 구현은 Fluentd입니다.
- 로그 저장소 - 로그가 저장되는 위치입니다. 기본 구현은 Elasticsearch입니다. 기본 Elasticsearch 로그 저장소를 사용하거나 외부 로그 저장소로 로그를 전달할 수 있습니다. 기본 로그 저장소는 단기 저장소에 대해 최적화되고 테스트되었습니다.
- 시각화 - 로그, 그래프, 차트 등을 보는 데 사용할 수 있는 UI 구성 요소입니다. 현재 구현은 Kibana입니다.

이 문서에서 이 문서는 언급된 경우를 제외하고 로그 저장소 또는 Elasticsearch, 시각화 또는 Kibana, 수집 또는 Fluentd를 상호 교환적으로 참조할 수 있습니다.

#### 1.1.2. 로깅 수집기 정보

OpenShift Container Platform은 Fluentd를 사용하여 컨테이너 및 노드 로그를 수집합니다.

기본적으로 로그 수집기는 다음 소스를 사용합니다.

- 모든 시스템 로그에 대한 `journald`
- 모든 컨테이너 로그에 대한 `/var/log/containers/*.log`

로깅 수집기는 각 OpenShift Container Platform 노드에 포드를 배포하는 DaemonSet로 배포됩니다. 시스템 및 인프라 로그는 운영 체제, 컨테이너 런타임 및 OpenShift 컨테이너 플랫폼에서 `journald`에 의해 로그 메시지로 생성됩니다. 애플리케이션 로그는 CRI-O 컨테이너 엔진에 의해 생성됩니다. Fluentd는 이러



한 소스에서 로그를 수집하여 OpenShift Container Platform에서 구성할 때 내부 또는 외부로 전달합니다.

컨테이너 런타임은 로그 메시지의 소스(프로젝트, 포드 이름 및 컨테이너 ID)를 식별하기 위한 최소한의 정보를 제공합니다. 이는 로그 소스를 고유하게 식별하기에 충분하지 않습니다. 로그 수집기에서 로그 처리를 시작하기 전에 지정된 이름과 프로젝트가 있는 포드를 삭제하면 레이블 및 주석과 같은 API 서버의 정보를 사용할 수 없습니다. 로그 메시지를 비슷한 이름의 포드와 구별하고 로그를 소스로 프로젝트하거나 추적하는 방법이 없을 수 있습니다. 이 제한은 로그 수집 및 정규화가 **최선의 노력**으로 간주됨을 의미합니다.



#### 중요

사용 가능한 컨테이너 런타임은 최소한의 정보를 제공하여 로그 메시지의 소스를 식별하고 고유한 개별 로그 메시지를 보장하지 않거나 이러한 메시지가 소스로 추적될 수 있음을 보증하지 않습니다.

### 1.1.3. 로그 저장소 정보

기본적으로 OpenShift Container Platform은 [ES\(Elasticsearch\)](#)를 사용하여 로그 데이터를 저장합니다. 선택적으로 로그 전달 기능을 사용하여 로그를 외부 로그 저장소로 전달하도록 Fluentd 프로토콜, syslog 프로토콜 또는 OpenShift Container Platform Log Forwarding API를 사용할 수 있습니다.

클러스터 로깅 Elasticsearch 인스턴스는 약 7일 동안 단기 저장을 위해 최적화 및 테스트되었습니다. 장기적으로 로그를 유지하려면 데이터를 타사 스토리지 시스템으로 이동하는 것이 좋습니다.

Elasticsearch는 Fluentd의 로그 데이터를 데이터 저장소 또는 인덱스로 구성된 다음 각 인덱스를 *shards*라고 하는 여러 조각으로 세분화합니다. 이 조각은 Elasticsearch 클러스터의 Elasticsearch 노드 세트에 분산됩니다. Elasticsearch가 Elasticsearch 노드에 분산되는 *복제본*이라는 shard의 사본을 작성하도록 Elasticsearch를 구성할 수 있습니다. 클러스터 로깅 사용자 정의 리소스(CR)를 사용하면 shard가 복제되는 방식을 지정하여 데이터 중복성과 장애에 대한 복원력을 제공할 수 있습니다. 클러스터 로깅 CR의 보존 정책을 사용하여 다른 유형의 로그를 보존하는 기간을 지정할 수도 있습니다.



#### 참고

인덱스 템플릿의 기본 shard 수는 Elasticsearch 데이터 노드 수와 같습니다.

클러스터 로깅 Operator 및 도우미 Elasticsearch Operator는 각 Elasticsearch 노드가 자체 스토리지 볼륨을 포함하는 고유한 배치를 사용하여 배치되도록 합니다. 필요에 따라 클러스터 로깅 사용자 정의 리소스(CR)를 사용하여 Elasticsearch 노드 수를 늘릴 수 있습니다. 스토리지 구성과 관련된 고려 사항은 [Elasticsearch 설명서](#)를 참조하십시오.



#### 참고

고가용성 Elasticsearch 환경에는 각각 서로 다른 호스트에 있는 최소 3개의 Elasticsearch 노드가 필요합니다.

Elasticsearch 인덱스에 적용된 RBAC(역할 기반 액세스 제어)를 사용하면 개발자에 대한 로그 액세스를 제어할 수 있습니다. 관리자는 모든 로그에 액세스할 수 있으며 개발자는 프로젝트의 로그에만 액세스할 수 있습니다.

### 1.1.4. 로깅 시각화 정보

OpenShift Container Platform은 Kibana를 사용하여 Fluentd에서 수집하고 Elasticsearch에서 인덱싱된 로그 데이터를 표시합니다.

Kibana는 히스토그램, 선 그래프, 원형 차트 및 기타 시각화를 통해 Elasticsearch 데이터를 쿼리, 검색 및 시각화할 수 있는 브라우저 기반 콘솔 인터페이스입니다.

### 1.1.5. 이벤트 라우팅 정보

이벤트 라우터는 클러스터 로깅으로 수집할 수 있도록 OpenShift Container Platform 이벤트를 감시하는 포드입니다. 이벤트 라우터는 모든 프로젝트에서 이벤트를 수집하여 **STDOUT**에 작성합니다. Fluentd는 이러한 이벤트를 수집하여 OpenShift Container Platform Elasticsearch 인스턴스로 전달합니다. Elasticsearch는 이벤트를 **인프라** 인덱스에 인덱싱합니다.

이벤트 라우터를 수동으로 배포해야 합니다.

## 2장. 클러스터 로깅 배포

Elasticsearch 및 Cluster Logging Operator를 배포하여 클러스터 로깅을 설치할 수 있습니다.

Elasticsearch Operator는 클러스터 로깅에 사용되는 Elasticsearch 클러스터를 생성 및 관리합니다. 클러스터 로깅 Operator는 SR-IOV 스택의 구성 요소를 생성하고 관리합니다.

OpenShift Container Platform에 클러스터 로깅을 배포하는 프로세스에는 다음이 포함됩니다.

- [클러스터 로깅 스토리지 고려 사항](#) 검토.
- OpenShift Container Platform [웹 콘솔](#) 또는 [CLI](#)를 사용하여 Elasticsearch Operator 및 Cluster Logging Operator 설치

### 2.1. 웹 콘솔을 사용하여 클러스터 로깅 설치

OpenShift Container Platform 웹 콘솔을 사용하여 Elasticsearch 및 클러스터 로깅 Operator를 설치할 수 있습니다.

#### 사전 요구 사항

- Elasticsearch에 필요한 영구 스토리지가 있는지 확인하십시오. 각 Elasticsearch 노드에는 자체 스토리지 볼륨이 필요합니다.  
Elasticsearch는 메모리를 많이 사용하는 애플리케이션입니다. 기본적으로 OpenShift Container Platform은 메모리 요청 및 제한이 16GB인 3 개의 Elasticsearch 노드를 설치합니다. 이 초기 3개의 OpenShift Container Platform 노드 세트에는 클러스터 내에서 Elasticsearch를 실행하기에 충분한 메모리가 없을 수 있습니다. Elasticsearch와 관련된 메모리 문제가 발생하는 경우 기존 노드의 메모리를 늘리는 대신 클러스터에 Elasticsearch 노드를 더 추가해야 합니다.

#### 프로시저

OpenShift Container Platform 웹 콘솔을 사용하여 Elasticsearch Operator 및 Cluster Logging Operator를 설치하려면 다음을 수행합니다.

1. Elasticsearch Operator를 설치합니다.
  - a. OpenShift Container Platform 웹 콘솔에서 **Operator** → **OperatorHub**를 클릭합니다.
  - b. 사용 가능한 Operator 목록에서 **Elasticsearch Operator**를 선택한 다음 **설치**를 클릭합니다.
  - c. **설치 모드**에서 **클러스터의 모든 네임스페이스**가 선택되어 있는지 확인하십시오.
  - d. **설치된 네임스페이스**에서 **openshift-operators-redhat**이 선택되어 있는지 확인하십시오.  
**openshift-operators-redhat** 네임스페이스를 지정해야 합니다. **openshift-operators** 네임스페이스에는 신뢰할 수 없는 커뮤니티 Operator가 포함될 수 있으며 OpenShift Container Platform 지표와 동일한 이름의 지표를 게시 할 수 있어 충돌이 발생할 수 있습니다.
  - e. 이 네임스페이스에서 **Operator** 권장 클러스터 모니터링 사용을 선택합니다.  
이 옵션은 네임스페이스 오브젝트에서 **openshift.io/cluster-monitoring: "true"** 레이블을 설정합니다. 클러스터 모니터링이 **openshift-operators-redhat** 네임스페이스를 스캔하도록 하려면 이 옵션을 선택해야 합니다.
  - f. **4.6을 업데이트 채널**로 선택합니다.
  - g. **승인 전략**을 선택합니다.

- 자동 전략을 사용하면 OLM(Operator 수명 주기 관리자)이 새 버전을 사용할 수 있을 때 Operator를 자동으로 업데이트할 수 있습니다.
  - 수동 전략을 사용하려면 적절한 자격 증명을 가진 사용자가 Operator 업데이트를 승인해야 합니다.
- h. 설치를 클릭합니다.
- i. **Operator** → **설치된 Operator** 페이지로 전환하여 Elasticsearch Operator가 설치되었는지 확인하십시오.
- j. **상태**가 **성공**인 모든 프로젝트에 **Elasticsearch Operator**가 나열되어 있는지 확인하십시오.
2. 클러스터 로깅 Operator를 설치합니다.
- a. OpenShift Container Platform 웹 콘솔에서 **Operator** → **OperatorHub**를 클릭합니다.
- b. 사용 가능한 Operator 목록에서 **클러스터 로깅**을 선택한 다음 **설치**를 클릭합니다.
- c. **설치 모드**에서 **클러스터의 특정 네임스페이스**가 선택되어 있는지 확인하십시오.
- d. **Operator** 권장 네임스페이스가 설치된 네임스페이스에서 **openshift-logging**인지 확인하십시오.
- e. 이 네임스페이스에서 **Operator** 권장 클러스터 모니터링 사용을 선택합니다.  
이 옵션은 네임스페이스 오브젝트에서 **openshift.io/cluster-monitoring: "true"** 레이블을 설정합니다. 클러스터 모니터링이 **openshift-logging** 네임스페이스를 스크랩하도록 하려면 이 옵션을 선택해야 합니다.
- f. **4.6**을 **업데이트 채널**로 선택합니다.
- g. **승인 전략**을 선택합니다.
- 자동 전략을 사용하면 OLM(Operator 수명 주기 관리자)이 새 버전을 사용할 수 있을 때 Operator를 자동으로 업데이트할 수 있습니다.
  - 수동 전략을 사용하려면 적절한 자격 증명을 가진 사용자가 Operator 업데이트를 승인해야 합니다.
- h. **설치**를 클릭합니다.
- i. **Operators** → **설치된 Operator** 페이지로 전환하여 클러스터 로깅 Operator가 설치되었는지 확인하십시오.
- j. 클러스터 로깅이 **openshift-logging** 프로젝트에 **성공 상태**로 나열되어 있는지 확인합니다. Operator가 설치된 것으로 나타나지 않으면 다음과 같이 추가 문제 해결을 수행하십시오.
- **Operators** → **설치된 Operator** 페이지로 전환하여 **상태** 열에 오류 또는 실패가 있는지 점검하십시오.
  - **워크로드** → **포드** 페이지로 전환하고 **openshift-logging** 프로젝트에서 문제를 보고하는 포드의 로그를 확인하십시오.
3. 클러스터 로깅 인스턴스를 생성합니다.
- a. **관리** → **사용자 정의 리소스 정의** 페이지로 전환합니다.
- b. **사용자 정의 리소스 정의** 페이지에서 **ClusterLogging**을 클릭합니다.

- c. 사용자 정의 리소스 정의 개요 페이지의 작업 메뉴에서 인스턴스 보기를 선택합니다.
- d. **ClusterLoggings** 페이지에서 **ClusterLogging** 생성을 클릭합니다.  
데이터를 로드하기 위해 페이지를 새로 고쳐야 할 수도 있습니다.
- e. YAML 필드에서 코드를 다음으로 바꿉니다.



### 참고

이 기본 클러스터 로깅 구성은 다양한 환경을 지원해야 합니다. 클러스터 로깅 클러스터에 수행할 수 있는 수정 사항에 대한 정보는 클러스터 로깅 구성 요소 튜닝 및 구성 주제를 검토합니다.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance" ❶
  namespace: "openshift-logging"
spec:
  managementState: "Managed" ❷
  logStore:
    type: "elasticsearch" ❸
    retentionPolicy: ❹
      application:
        maxAge: 1d
      infra:
        maxAge: 7d
      audit:
        maxAge: 7d
    elasticsearch:
      nodeCount: 3 ❺
      storage:
        storageClassName: "<storage-class-name>" ❻
        size: 200G
      redundancyPolicy: "SingleRedundancy"
  visualization:
    type: "kibana" ❼
    kibana:
      replicas: 1
  curation:
    type: "curator"
    curator:
      schedule: "30 3 * * *" ❽
  collection:
    logs:
      type: "fluentd" ❾
      fluentd: {}
```

- ❶ 이름은 **instance** 이어야 합니다.
- ❷ 클러스터 로깅 관리 상태. 경우에 따라 클러스터 로깅 기본값을 변경하는 경우 이를 **관리되지 않음**으로 설정해야 합니다. 그러나 관리되지 않는 배포는 클러스터 로깅이 다시 관리 상태로 될 때까지 업데이트를 받지 않습니다.

- 3 Elasticsearch 구성을 위한 설정입니다. CR을 사용하여 shard 복제 정책 및 영구 스토리지를 구성할 수 있습니다.
- 4 Elasticsearch가 각 로그 소스를 유지해야 하는 시간을 지정합니다. 정수 및 시간 지정을 입력합니다(주(w), 시간(h/H), 분(m) 및 초(s)). 예를 들어 7일은 **7d**입니다. **maxAge**보다 오래된 로그는 삭제됩니다. 각 로그 소스에 대한 보존 정책을 지정해야 합니다. 그렇지 않으면 해당 소스에 대해 Elasticsearch 인덱스가 생성되지 않습니다.
- 5 Elasticsearch 노드 수를 지정합니다. 이 목록 뒤에 나오는 참고 사항을 참조하십시오.
- 6 Elasticsearch 스토리지의 기존 StorageClass 이름을 입력합니다. 최상의 성능을 위해서는 블록 스토리지를 할당하는 StorageClass를 지정합니다. StorageClass를 지정하지 않으면 OpenShift Container Platform은 임시 스토리지로만 클러스터 로깅을 배포합니다.
- 7 Kibana 구성을 위한 설정입니다. CR을 사용하여 중복성을 위해 Kibana를 확장하고 Kibana 노드의 CPU 및 메모리를 구성할 수 있습니다. 자세한 내용은 **Kibana 구성**을 참조하십시오.
- 8 Curator 일정 구성을 위한 설정입니다. Curator는 OpenShift Container Platform 4.5 이전의 Elasticsearch 인덱스 형식의 데이터를 삭제하는 데 사용되며 이후 릴리스에서 제거됩니다.
- 9 Fluentd 구성을 위한 설정입니다. CR을 사용하여 Fluentd CPU 및 메모리 제한을 구성할 수 있습니다. 자세한 내용은 **Fluentd 구성**을 참조하십시오.



## 참고

Elasticsearch 마스터 노드의 최대 수는 3입니다. **3**보다 큰 **nodeCount**를 지정하면 OpenShift Container Platform은 마스터, 클라이언트 및 데이터 역할을 가진 마스터 적격 노드인 Elasticsearch 노드 3개를 생성합니다. 추가 Elasticsearch 노드는 클라이언트 및 데이터 역할을 사용하여 데이터 전용 노드로 생성됩니다. 마스터 노드는 인덱스 작성 또는 삭제, shard 할당 및 추적 노드와 같은 클러스터 전체 작업을 수행합니다. 데이터 노드는 shard를 보유하고 CRUD, 검색 및 집계와 같은 데이터 관련 작업을 수행합니다. 데이터 관련 작업은 I/O, 메모리 및 CPU 집약적입니다. 현재 노드에 과부하가 걸리면 이러한 리소스를 모니터링하고 더 많은 데이터 노드를 추가하는 것이 중요합니다.

예를 들어 **nodeCount = 4**인 경우 다음 노드가 생성됩니다.

```
$ oc get deployment
```

## 출력 예

```
cluster-logging-operator    1/1    1    1    18h
elasticsearch-cd-x6kdekli-1  0/1    1    0    6m54s
elasticsearch-cdm-x6kdekli-1 1/1    1    1    18h
elasticsearch-cdm-x6kdekli-2  0/1    1    0    6m49s
elasticsearch-cdm-x6kdekli-3  0/1    1    0    6m44s
```

인덱스 템플릿의 기본 shard 수는 Elasticsearch 데이터 노드 수와 같습니다.

- f. **생성**을 클릭합니다. 이렇게 하면 클러스터 로깅 구성 요소, Elasticsearch 사용자 정의 리소스 및 구성 요소, Kibana 인터페이스가 생성됩니다.

4. 설치를 확인하십시오.

a. 워크로드 → 포드 페이지로 전환합니다.

b. **openshift-logging** 프로젝트를 선택합니다.

다음 목록과 유사한 클러스터 로깅, Elasticsearch, Fluentd 및 Kibana에 대한 여러 포드가 표시됩니다.

- cluster-logging-operator-cb795f8dc-xkckc
- elasticsearch-cdm-b3nqzchd-1-5c6797-67kfz
- elasticsearch-cdm-b3nqzchd-2-6657f4-wtprv
- elasticsearch-cdm-b3nqzchd-3-588c65-clg7g
- fluentd-2c7dg
- fluentd-9z7kk
- fluentd-br7r2
- fluentd-fn2sb
- fluentd-pb2f8
- fluentd-zqgqx
- kibana-7fb4fd4cc9-bvt4p

## 2.2. CLI를 사용하여 클러스터 로깅 설치

OpenShift Container Platform CLI를 사용하여 Elasticsearch 및 클러스터 로깅 Operator를 설치할 수 있습니다.

### 사전 요구 사항

- Elasticsearch에 필요한 영구 스토리지가 있는지 확인하십시오. 각 Elasticsearch 노드에는 자체 스토리지 볼륨이 필요합니다.

Elasticsearch는 메모리를 많이 사용하는 애플리케이션입니다. 기본적으로 OpenShift Container Platform은 메모리 요청 및 제한이 16GB인 3 개의 Elasticsearch 노드를 설치합니다. 이 초기 3개의 OpenShift Container Platform 노드 세트에는 클러스터 내에서 Elasticsearch를 실행하기에 충분한 메모리가 없을 수 있습니다. Elasticsearch와 관련된 메모리 문제가 발생하는 경우 기존 노드의 메모리를 늘리는 대신 클러스터에 Elasticsearch 노드를 더 추가해야 합니다.

### 프로시저

CLI를 사용하여 Elasticsearch Operator 및 클러스터 로깅 Operator를 설치하려면 다음을 수행합니다.

1. Elasticsearch Operator의 네임스페이스를 만듭니다.

a. Elasticsearch Operator에 대한 네임스페이스 오브젝트 YAML 파일(예: **eo-namespace.yaml**)을 생성합니다.

```
apiVersion: v1
kind: Namespace
metadata:
  name: openshift-operators-redhat 1
```

```

annotations:
  openshift.io/node-selector: ""
labels:
  openshift.io/cluster-monitoring: "true" 2

```

- 1 **openshift-operators-redhat** 네임스페이스를 지정해야 합니다. 지표와의 충돌을 방지하려면 **openshift-operators** 네임스페이스가 아니라 **openshift-operators-redhat** 네임스페이스에서 지표를 스크랩하도록 Prometheus 클러스터 모니터링 스택을 구성해야 합니다. **openshift-operators** 네임스페이스에는 신뢰할 수 없는 커뮤니티 Operator가 포함될 수 있으며 OpenShift Container Platform 지표와 동일한 이름의 지표를 게시할 수 있어 충돌이 발생할 수 있습니다.
- 2 클러스터 모니터링이 **openshift-operators-redhat** 네임스페이스를 스크랩하도록 하려면 이 레이블을 지정해야 합니다.

b. 네임스페이스를 생성합니다.

```
$ oc create -f <file-name>.yaml
```

예를 들면 다음과 같습니다.

```
$ oc create -f eo-namespace.yaml
```

2. 클러스터 로깅 Operator의 네임스페이스를 만듭니다.

- a. 클러스터 로깅 Operator에 대한 네임스페이스 오브젝트 YAML 파일(예: **clo-namespace.yaml**)을 생성합니다.

```

apiVersion: v1
kind: Namespace
metadata:
  name: openshift-logging
  annotations:
    openshift.io/node-selector: ""
  labels:
    openshift.io/cluster-monitoring: "true"

```

b. 네임스페이스를 생성합니다.

```
$ oc create -f <file-name>.yaml
```

예를 들면 다음과 같습니다.

```
$ oc create -f clo-namespace.yaml
```

3. 다음 오브젝트를 만들어 Elasticsearch Operator를 설치합니다.

- a. Elasticsearch Operator에 대한 Operator 그룹 오브젝트 YAML 파일(예: **eo-og.yaml**)을 생성합니다.

```

apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:

```



```
name: openshift-operators-redhat
namespace: openshift-operators-redhat ❶
spec: {}
```

- ❶ **openshift-operators-redhat** 네임스페이스를 지정해야 합니다.

- b. Operator 그룹 오브젝트를 생성합니다.

```
$ oc create -f <file-name>.yaml
```

예를 들면 다음과 같습니다.

```
$ oc create -f eo-og.yaml
```

- c. 서브스크립션 오브젝트 YAML 파일(예: **eo-sub.yaml**)을 만들어 Elasticsearch Operator에 네임스페이스를 서브스크립션합니다.

#### 서브스크립션 예

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: "elasticsearch-operator"
  namespace: "openshift-operators-redhat" ❶
spec:
  channel: "4.6" ❷
  installPlanApproval: "Automatic"
  source: "redhat-operators" ❸
  sourceNamespace: "openshift-marketplace"
  name: "elasticsearch-operator"
```

- ❶ **openshift-operators-redhat** 네임스페이스를 지정해야 합니다.

- ❷ **4.6** 을 채널로 지정합니다.

- ❸ **redhat-operators**를 지정합니다. OpenShift Container Platform 클러스터가 연결이 끊긴 클러스터라고도 하는 제한된 네트워크에 설치된 경우 OLM(Operator Lifecycle Manager)을 구성할 때 생성된 CatalogSource 오브젝트의 이름을 지정합니다.

- d. 서브스크립션 오브젝트를 생성합니다.

```
$ oc create -f <file-name>.yaml
```

예를 들면 다음과 같습니다.

```
$ oc create -f eo-sub.yaml
```

Elasticsearch Operator는 **openshift-operators-redhat** 네임스페이스에 설치되고 클러스터의 각 프로젝트에 복사됩니다.

- e. Operator 설치를 확인하십시오.

```
oc get csv --all-namespaces
```

#### 출력 예

NAMESPACE	NAME	DISPLAY
VERSION	REPLACES	PHASE
default	elasticsearch-operator.4.6.0-202007012112.p0	
Elasticsearch Operator	4.6.0-202007012112.p0	Succeeded
kube-node-lease	elasticsearch-operator.4.6.0-202007012112.p0	
Elasticsearch Operator	4.6.0-202007012112.p0	Succeeded
kube-public	elasticsearch-operator.4.6.0-202007012112.p0	
Elasticsearch Operator	4.6.0-202007012112.p0	Succeeded
kube-system	elasticsearch-operator.4.6.0-202007012112.p0	
Elasticsearch Operator	4.6.0-202007012112.p0	Succeeded
openshift-apiserver-operator	elasticsearch-operator.4.6.0-202007012112.p0	
202007012112.p0	Elasticsearch Operator	4.6.0-202007012112.p0
Succeeded		
openshift-apiserver	elasticsearch-operator.4.6.0-202007012112.p0	
Elasticsearch Operator	4.6.0-202007012112.p0	Succeeded
openshift-authentication-operator	elasticsearch-operator.4.6.0-202007012112.p0	
202007012112.p0	Elasticsearch Operator	4.6.0-202007012112.p0
Succeeded		
openshift-authentication	elasticsearch-operator.4.6.0-202007012112.p0	
202007012112.p0	Elasticsearch Operator	4.6.0-202007012112.p0
Succeeded		
...		

각 네임스페이스에는 Elasticsearch Operator가 있어야 합니다. 버전 번호가 표시된 것과 다를 수 있습니다.

#### 4. 다음 오브젝트를 생성하여 클러스터 로깅 Operator를 설치합니다.

- 클러스터 로깅 Operator에 대한 OperatorGroup 오브젝트 YAML 파일(예: **clo-og.yaml**)을 생성합니다.

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: cluster-logging
  namespace: openshift-logging ❶
spec:
  targetNamespaces:
    - openshift-logging ❷
```

❶ ❷ **openshift-logging** 네임스페이스를 지정해야 합니다.

- OperatorGroup 오브젝트를 생성합니다.

```
$ oc create -f <file-name>.yaml
```

예를 들면 다음과 같습니다.

```
$ oc create -f clo-og.yaml
```

- c. 서브스크립션 오브젝트 YAML 파일(예: **clo-sub.yaml**)을 생성하여 클러스터 로깅 Operator에 네임스페이스를 서브스크립션합니다.

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: cluster-logging
  namespace: openshift-logging ❶
spec:
  channel: "4.6" ❷
  name: cluster-logging
  source: redhat-operators ❸
  sourceNamespace: openshift-marketplace
```

- ❶ **openshift-logging** 네임스페이스를 지정해야 합니다.
- ❷ **4.6**을 채널로 지정합니다.
- ❸ **redhat-operators**를 지정합니다. OpenShift Container Platform 클러스터가 연결이 끊긴 클러스터라고도 하는 제한된 네트워크에 설치된 경우 OLM(Operator Lifecycle Manager)을 구성할 때 생성된 CatalogSource 오브젝트의 이름을 지정합니다.

- d. 서브스크립션 오브젝트를 생성합니다.

```
$ oc create -f <file-name>.yaml
```

예를 들면 다음과 같습니다.

```
$ oc create -f clo-sub.yaml
```

클러스터 로깅 Operator는 **openshift-logging** 네임스페이스에 설치됩니다.

- e. Operator 설치를 확인하십시오.

**openshift-logging** 네임스페이스에 클러스터 로깅 Operator가 있어야 합니다. 버전 번호가 표시된 것과 다를 수 있습니다.

```
oc get csv -n openshift-logging
```

출력 예

NAMESPACE	NAME	DISPLAY
VERSION	REPLACES	PHASE
...		
openshift-logging	clusterlogging.4.6.0-202007012112.p0	
Cluster Logging	4.6.0-202007012112.p0	Succeeded
...		

5. 클러스터 로깅 인스턴스를 생성합니다.

- a. 클러스터 로깅 Operator에 대한 인스턴스 오브젝트 YAML 파일(예: **clo-instance.yaml**)을 생성합니다.



## 참고

이 기본 클러스터 로깅 구성은 다양한 환경을 지원해야 합니다. 클러스터 로깅 클러스터에 수행할 수 있는 수정 사항에 대한 정보는 클러스터 로깅 구성 요소 튜닝 및 구성 주제를 검토합니다.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance" ❶
  namespace: "openshift-logging"
spec:
  managementState: "Managed" ❷
  logStore:
    type: "elasticsearch" ❸
    retentionPolicy: ❹
      application:
        maxAge: 1d
      infra:
        maxAge: 7d
      audit:
        maxAge: 7d
    elasticsearch:
      nodeCount: 3 ❺
      storage:
        storageClassName: "<storage-class-name>" ❻
        size: 200G
        redundancyPolicy: "SingleRedundancy"
    visualization:
      type: "kibana" ❼
      kibana:
        replicas: 1
    curation:
      type: "curator"
      curator:
        schedule: "30 3 * * *" ❽
    collection:
      logs:
        type: "fluentd" ❾
        fluentd: {}
```

- ❶ 이름은 **instance** 이어야 합니다.
- ❷ 클러스터 로깅 관리 상태. 경우에 따라 클러스터 로깅 기본값을 변경하는 경우 이를 **관리되지 않음**으로 설정해야 합니다. 그러나 관리되지 않는 배포는 클러스터 로깅이 다시 관리 상태로 될 때까지 업데이트를 받지 않습니다. 배포를 다시 관리 상태로 설정하면 수정한 내용을 되돌릴 수 있습니다.
- ❸ Elasticsearch 구성을 위한 설정입니다. CR(사용자 정의 리소스)을 사용하여 shard 복제 정책 및 영구 스토리지를 구성할 수 있습니다.
- ❹ Elasticsearch가 각 로그 소스를 유지해야 하는 시간을 지정합니다. 정수 및 시간 지정을 입력합니다(주(w), 시간(h/H), 분(m) 및 초(s)). 예를 들어 7일은 **7d**입니다. **maxAge**보다 오래된 로그는 삭제됩니다. 각 로그 소스에 대한 보존 정책을 지정해야 합니다. 그렇

지 않으면 해당 소스에 대해 Elasticsearch 인덱스가 생성되지 않습니다.

- 5 Elasticsearch 노드 수를 지정합니다. 이 목록 뒤에 나오는 참고 사항을 참조하십시오.
- 6 Elasticsearch 스토리지의 기존 StorageClass 이름을 입력합니다. 최상의 성능을 위해서는 블록 스토리지를 할당하는 StorageClass를 지정합니다. StorageClass를 지정하지 않으면 OpenShift Container Platform은 임시 스토리지로만 클러스터 로깅을 배포합니다.
- 7 Kibana 구성을 위한 설정입니다. CR을 사용하여 중복성을 위해 Kibana를 확장하고 Kibana 노드의 CPU 및 메모리를 구성할 수 있습니다. 자세한 내용은 **Kibana 구성**을 참조하십시오.
- 8 Curator 일정 구성을 위한 설정입니다. Curator는 OpenShift Container Platform 4.5 이전의 Elasticsearch 인덱스 형식의 데이터를 삭제하는 데 사용되며 이후 릴리스에서 제거됩니다.
- 9 Fluentd 구성을 위한 설정입니다. CR을 사용하여 Fluentd CPU 및 메모리 제한을 구성할 수 있습니다. 자세한 내용은 **Fluentd 구성**을 참조하십시오.



## 참고

Elasticsearch 마스터 노드의 최대 수는 3입니다. 3보다 큰 **nodeCount**를 지정하면 OpenShift Container Platform은 마스터, 클라이언트 및 데이터 역할을 가진 마스터 적격 노드인 Elasticsearch 노드 3개를 생성합니다. 추가 Elasticsearch 노드는 클라이언트 및 데이터 역할을 사용하여 데이터 전용 노드로 생성됩니다. 마스터 노드는 인덱스 작성 또는 삭제, shard 할당 및 추적 노드와 같은 클러스터 전체 작업을 수행합니다. 데이터 노드는 shard를 보유하고 CRUD, 검색 및 집계와 같은 데이터 관련 작업을 수행합니다. 데이터 관련 작업은 I/O, 메모리 및 CPU 집약적입니다. 현재 노드에 과부하가 걸리면 이러한 리소스를 모니터링하고 더 많은 데이터 노드를 추가하는 것이 중요합니다.

예를 들어 **nodeCount = 4**인 경우 다음 노드가 생성됩니다.

```
$ oc get deployment
```

## 출력 예

```
cluster-logging-operator 1/1 1 1 18h
elasticsearch-cd-x6kdekli-1 1/1 1 0 6m54s
elasticsearch-cdm-x6kdekli-1 1/1 1 1 18h
elasticsearch-cdm-x6kdekli-2 1/1 1 0 6m49s
elasticsearch-cdm-x6kdekli-3 1/1 1 0 6m44s
```

인덱스 템플릿의 기본 shard 수는 Elasticsearch 데이터 노드 수와 같습니다.

b. 인스턴스를 만듭니다.

```
$ oc create -f <file-name>.yaml
```

예를 들면 다음과 같습니다.

```
$ oc create -f clo-instance.yaml
```

이렇게하면 클러스터 로깅 구성 요소, Elasticsearch 사용자 정의 리소스 및 구성 요소, Kibana 인터페이스가 생성됩니다.

6. **openshift-logging** 프로젝트에 포드를 나열하여 설치를 확인하십시오.

다음 목록과 유사한 클러스터 로깅, Elasticsearch, Fluentd 및 Kibana에 대한 여러 포드가 표시됩니다.

```
oc get pods -n openshift-logging
```

출력 예

NAME	READY	STATUS	RESTARTS	AGE
cluster-logging-operator-66f77fccb-ppzbg	1/1	Running	0	7m
elasticsearch-cdm-ftuhduuw-1-ffc4b9566-q6bhp	2/2	Running	0	2m40s
elasticsearch-cdm-ftuhduuw-2-7b4994dbfc-rd2gc	2/2	Running	0	2m36s
elasticsearch-cdm-ftuhduuw-3-84b5ff7ff8-gqnm2	2/2	Running	0	2m4s
fluentd-587vb	1/1	Running	0	2m26s
fluentd-7mpb9	1/1	Running	0	2m30s
fluentd-flm6j	1/1	Running	0	2m33s
fluentd-gn4rn	1/1	Running	0	2m26s
fluentd-nlgb6	1/1	Running	0	2m30s
fluentd-snpkt	1/1	Running	0	2m28s
kibana-d6d5668c5-rppqm	2/2	Running	0	2m39s

## 2.3. 추가 리소스

- Operator 설치에 대한 자세한 내용은 [OperatorHub](#)에서 [Operator 설치](#)를 참조하십시오.

## 3장. 클러스터 로깅 배포 구성

### 3.1. 클러스터 로깅 사용자 정의 리소스 정보

OpenShift Container Platform 클러스터 로깅을 구성하려면 클러스터 로깅 사용자 정의 리소스(CR)를 사용자 정의합니다.

#### 3.1.1. 클러스터 로깅 사용자 정의 리소스 정보

클러스터 로깅 배포를 변경하려면 클러스터 로깅 사용자 정의 리소스(CR)를 생성하고 수정합니다. CR을 작성하거나 수정하기 위한 지침이 이 문서에 적절하게 제공됩니다.

다음은 클러스터 로깅을 위한 일반적인 사용자 정의 리소스의 예입니다.

#### 샘플 클러스터 로깅 CR

```
kind: "ClusterLogging"
metadata:
  name: "instance" ❶
  namespace: "openshift-logging" ❷
spec:
  managementState: "Managed" ❸
  logStore:
    type: "elasticsearch" ❹
    retentionPolicy:
      application:
        maxAge: 1d
      infra:
        maxAge: 7d
      audit:
        maxAge: 7d
    elasticsearch:
      nodeCount: 3
      resources:
        limits:
          memory: 16Gi
        requests:
          cpu: 500m
          memory: 16Gi
      storage:
        storageClassName: "gp2"
        size: "200G"
      redundancyPolicy: "SingleRedundancy"
  visualization: ❺
    type: "kibana"
    kibana:
      resources:
        limits:
          memory: 736Mi
        requests:
          cpu: 100m
          memory: 736Mi
      replicas: 1
  curation: ❻
```

```

type: "curator"
curator:
  resources:
    limits:
      memory: 256Mi
    requests:
      cpu: 100m
      memory: 256Mi
  schedule: "30 3 * * *"
collection: 7
logs:
  type: "fluentd"
  fluentd:
    resources:
      limits:
        memory: 736Mi
      requests:
        cpu: 100m
        memory: 736Mi

```

- 1 CR 이름은 **instance**이어야 합니다.
- 2 CR은 **openshift-logging** 네임스페이스에 설치해야 합니다.
- 3 클러스터 로깅 Operator 관리 상태. **비 관리**로 설정된 경우 Operator는 지원되지 않는 상태이며 업데이트를 받지 않습니다.
- 4 보존 정책, 노드 수, 리소스 요청 및 제한, 스토리지 클래스를 포함한 로그 저장소 설정.
- 5 리소스 요청 및 제한, 포드 복제본 수를 포함한 시각화 도우미 설정.
- 6 리소스 요청 및 제한, 큐레이션 일정을 포함한 큐레이션 설정.
- 7 리소스 요청 및 제한을 포함한 로그 수집기 설정.

## 3.2. 로깅 수집기 구성

OpenShift Container Platform은 Fluentd를 사용하여 클러스터에서 작업 및 애플리케이션 로그를 수집하고 Kubernetes Pod 및 네임스페이스 메타 데이터로 데이터를 보장합니다.

로그 수집기의 CPU 및 메모리 제한을 구성하고 **로그 수집기 포드를 특정 노드로 이동** 할 수 있습니다. 클러스터 로깅 사용자 정의 리소스 CR의 **spec.collection.log.fluentd** 스탠자를 통해 로그 수집기에 대해 지원되는 모든 수정을 수행할 수 있습니다.

### 3.2.1. 지원되지 않는 구성 정보

지원되는 클러스터 로깅 구성 방법은 이 설명서에 설명된 옵션을 사용하여 구성하는 것입니다. 다른 구성은 지원되지 않으므로 사용하지 마십시오. 구성 패러다임은 OpenShift Container Platform 릴리스마다 변경될 수 있으며 이러한 경우는 모든 구성 가능성이 제어되는 경우에만 정상적으로 처리될 수 있습니다. 이 문서에 설명된 것과 다른 구성을 사용하는 경우 Elasticsearch Operator와 Cluster Logging Operator가 차이를 조정하므로 변경 사항이 사라집니다. Operator는 기본적으로 그리고 의도적으로 모든 항목을 정의된 상태로 되돌립니다.





## 참고

OpenShift Container Platform 설명서에 설명되지 않은 구성을 수행해야 하는 경우 클러스터 로깅 Operator 또는 Elasticsearch Operator를 관리되지 않음으로 설정해야 합니다. 관리되지 않는 클러스터 로깅 환경은 지원되지 않으며 클러스터 로깅을 관리 대상으로 반환할 때까지 업데이트를 받지 않습니다.

### 3.2.2. 로깅 수집기 포드 보기

**oc get pods --all-namespaces -o wide** 명령을 사용하여 Fluentd가 배포된 노드를 볼 수 있습니다.

#### 프로시저

**openshift-logging** 프로젝트에서 다음 명령을 실행합니다.

```
$oc get pods --selector component=fluentd -o wide -n openshift-logging
```

#### 출력 예

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED
fluentd-8d69v	1/1	Running	0	134m	10.130.2.30	master1.example.com	<none>
fluentd-bd225	1/1	Running	0	134m	10.131.1.11	master2.example.com	<none>
fluentd-cvrzs	1/1	Running	0	134m	10.130.0.21	master3.example.com	<none>
fluentd-gpqg2	1/1	Running	0	134m	10.128.2.27	worker1.example.com	<none>
fluentd-l9j7j	1/1	Running	0	134m	10.129.2.31	worker2.example.com	<none>

### 3.2.3. 로그 수집기 CPU 및 메모리 제한 구성

로그 수집기는 CPU 및 메모리 제한을 모두 조정할 수 있습니다.

#### 프로시저

1. **openshift-logging** 프로젝트에서 클러스터 로깅 사용자 정의 리소스(CR)를 편집합니다.

```
$ oc edit ClusterLogging instance
```

```
$ oc edit ClusterLogging instance
```

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
```

```
....
```

```
spec:
  collection:
    logs:
```

```

fluentd:
  resources:
    limits: ❶
    memory: 736Mi
  requests:
    cpu: 100m
    memory: 736Mi

```

- ❶ 필요에 따라 CPU 및 메모리 제한 및 요청을 지정합니다. 표시된 값이 기본값입니다.

### 3.2.4. 로그 전달자를 위한 고급 구성

클러스터 로깅에는 Fluentd 로그 전달자의 성능을 조정하는 데 사용할 수 있는 여러 Fluentd 매개 변수가 포함됩니다. 이러한 매개변수를 사용하여 다음 Fluentd 동작을 변경할 수 있습니다.

- Fluentd 청크 및 청크 버퍼의 크기
- Fluentd 청크 플러싱 동작
- Fluentd 청크 전달 재시도 동작

Fluentd는 **청크**라는 단일 blob에서 로그 데이터를 수집합니다. Fluentd가 청크를 생성할 때 청크는 **스태이**지에 있는 것으로 간주되어 청크가 데이터로 채워집니다. 청크가 가득 차면 Fluentd는 청크를 **큐**로 이동합니다. 여기서 청크는 플러시되기 전에 보관되거나 대상에 기록됩니다. Fluentd는 네트워크 문제 또는 대상의 용량 문제와 같은 여러 가지 이유로 청크를 플러시하지 못할 수 있습니다. 청크를 플러시할 수 없는 경우 Fluentd는 구성된대로 플러시를 다시 시도합니다.

기본적으로 OpenShift Container Platform에서 Fluentd는 **지수 백오프** 방법을 사용하여 플러시를 다시 시도합니다. 여기서 Fluentd는 다시 플러시를 다시 시도하는 사이에 대기하는 시간을 두 배로 늘려 대상에 대한 연결 요청을 줄이는 데 도움이 됩니다. 지수 백오프를 비활성화하고 대신 **주기적 재시도** 방법을 사용하여 지정된 간격으로 청크 플러시를 재시도 할 수 있습니다. 기본적으로 Fluentd는 청크 플러싱을 무기한 재 시도합니다. OpenShift Container Platform에서는 무제한 재시도 동작을 변경할 수 없습니다.

이러한 매개변수는 대기 시간과 처리량 간의 균형을 결정하는 데 도움이 될 수 있습니다.

- 처리량에 대해 Fluentd를 최적화하려면 이러한 매개변수를 사용하여 더 큰 버퍼 및 큐를 구성하고, 플러시를 지연하고, 재시도 간격을 더 길게 설정하여 네트워크 패킷 수를 줄일 수 있습니다. 버퍼가 클수록 노드 파일 시스템에 더 많은 공간이 필요합니다.
- 짧은 지연 시간을 최적화하기 위해 매개변수를 사용하여 가능한 한 빨리 데이터를 전송하고, 배치 축적을 방지하고, 큐와 버퍼를 더 짧게 만들고, 더 자주 플러시 및 재시도를 사용할 수 있습니다.

클러스터 로깅 사용자 정의 리소스(CR)에서 다음 매개변수를 사용하여 청크 및 플러시 동작을 구성할 수 있습니다. 그런 다음 Fluentd에서 사용할 수 있도록 매개변수가 Fluentd ConfigMap에 자동으로 추가됩니다.

## 참고

이러한 매개변수는 다음과 같습니다.

- 대부분의 사용자와 관련이 없습니다. 기본 설정은 좋은 일반 성능을 제공해야 합니다.
- Fluentd 구성 및 성능에 대한 자세한 지식이 있는 고급 사용자에게만 해당됩니다.
- 성능 조정 전용입니다. 로깅의 기능적 측면에는 영향을 미치지 않습니다.

표 3.1. 고급 Fluentd 구성 매개변수

매개변수	설명	기본
<b>chunkLimitSize</b>	각 청크의 최대 크기입니다. Fluentd는 이 크기에 도달하면 청크에 데이터 쓰기를 중지합니다. 그런 다음 Fluentd는 청크를 큐로 보내고 새 청크를 엽니다.	<b>8m</b>
<b>totalLimitSize</b>	스테이지와 큐의 총 크기인 버퍼의 최대 크기입니다. 버퍼 크기가 이 값을 초과하면 Fluentd는 청크로의 데이터 추가를 중지하고 오류와 함께 실패합니다. 청크에 없는 모든 데이터는 손실됩니다.	<b>8G</b>
<b>flushInterval</b>	청크 플러시 간격입니다. <b>s</b> (초), <b>m</b> (분), <b>h</b> (시간) 또는 <b>d</b> (일)를 사용할 수 있습니다.	<b>1s</b>
<b>flushMode</b>	플러시를 수행하는 방법: <ul style="list-style-type: none"> <li>● <b>lazy: timekey</b> 매개변수를 기반으로 청크를 플러시합니다. <b>timekey</b> 매개변수는 수정할 수 없습니다.</li> <li>● <b>interval: flushInterval</b> 매개변수를 기반으로 청크를 플러시합니다.</li> <li>● <b>immediate</b>: 데이터가 청크에 추가된 직후 청크를 플러시합니다.</li> </ul>	간격
<b>flushThreadCount</b>	청크 플러시를 수행하는 스레드 수입니다. 스레드 수를 늘리면 플러시 처리량이 향상되어 네트워크 대기 시간이 숨겨집니다.	<b>2</b>

매개변수	설명	기본
<b>overflowAction</b>	<p>큐가 가득 찼을 때 청크 동작:</p> <ul style="list-style-type: none"> <li>● <b>throw_exception</b>: 로그에 표시할 예외를 발생시킵니다.</li> <li>● <b>block</b>: 전체 버퍼 문제가 해결될 때까지 데이터 청크를 중지합니다.</li> <li>● <b>drop_oldest_chunk</b>: 새로 들어오는 청크를 수락하기 위해 가장 오래된 청크를 삭제합니다. 오래된 청크는 새로운 청크보다 가치가 적습니다.</li> </ul>	블록
<b>queuedChunkLimitSize</b>	큐의 청크 수입니다.	32
<b>retryMaxInterval</b>	<b>exponential_backoff</b> 재시도 방법의 최대 시간(초)입니다.	300s
<b>retryType</b>	<p>플러시 실패 시 재시도 방법:</p> <ul style="list-style-type: none"> <li>● <b>exponential_backoff</b>: 플러시 재시도 사이의 시간을 늘립니다. Fluentd는 <b>retry_max_interval</b> 매개변수에 도달할 때까지 다음 재시도까지 대기하는 시간을 두 배로 늘립니다.</li> <li>● <b>periodic: retryWait</b> 매개변수를 기반으로 주기적으로 플러시를 재시도합니다.</li> </ul>	<b>exponential_backoff</b>
<b>retryWait</b>	다음 청크 플러시 전의 시간(초)입니다.	1s

Fluentd 청크 수명 주기에 대한 자세한 내용은 Fluentd 문서의 [버퍼 플러그인](#)을 참조하십시오.

## 프로시저

1. **openshift-logging** 프로젝트에서 클러스터 로깅 사용자 정의 리소스(CR)를 편집합니다.

```
$ oc edit ClusterLogging instance
```

2. 다음 매개변수를 추가하거나 수정합니다.

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
  name: instance
  namespace: openshift-logging
spec:
  forwarder:
    fluentd:
      buffer:
        chunkLimitSize: 8m ❶
        flushInterval: 5s ❷
        flushMode: interval ❸
        flushThreadCount: 3 ❹
        overflowAction: throw_exception ❺
        retryMaxInterval: "300s" ❻
        retryType: periodic ❼
        retryWait: 1s ❽
        totalLimitSize: 32m ❾
      ...
```

- ❶ 플러시를 위해 큐에 추가되기 전에 각 청크의 최대 크기를 지정합니다.
- ❷ 청크 플러시 간격을 지정합니다.
- ❸ 청크 플러시를 수행할 수 있는 방법을 지정합니다: **lazy**, **interval** 또는 **immediate**.
- ❹ 청크 플러시에 사용할 스레드 수를 지정합니다.
- ❺ 큐가 가득 찼을 때 청크 동작을 지정합니다: **throw\_exception**, **block** 또는 **drop\_oldest\_chunk**.
- ❻ **exponential\_backoff** 청크 플러시 방법의 최대 간격(초)을 지정합니다.
- ❼ 청크 플러시 실패 시 재시도 유형을 **exponential\_backoff** 또는 **periodic**으로 지정합니다.
- ❽ 다음 청크 플러시 전 시간(초)을 지정합니다.
- ❾ 청크 버퍼의 최대 크기를 지정합니다.

3. Fluentd 포트가 재배포되었는지 확인합니다.

```
$ oc get pods -n openshift-logging
```

4. 새 값이 **fluentd** ConfigMap에 있는지 확인합니다.

```
$ oc extract configmap/fluentd --confirm
```

예: **fluentd.conf**

```
<buffer>
@type file
```

```

path '/var/lib/fluentd/default'
flush_mode interval
flush_interval 5s
flush_thread_count 3
flush_at_shutdown true
retry_type periodic
retry_wait 1s
retry_max_interval 300s
retry_forever true
queued_chunks_limit_size "#{ENV['BUFFER_QUEUE_LIMIT'] || '32'}"
total_limit_size "#{ENV['TOTAL_LIMIT_SIZE'] || 8589934592}" #32M
chunk_limit_size "#{ENV['BUFFER_SIZE_LIMIT'] || '8m'}"
overflow_action throw_exception
</buffer>

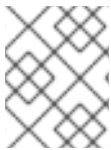
```

### 3.3. 로그 저장소 구성

OpenShift Container Platform은 Elasticsearch 6(ES)을 사용하여 로그 데이터를 저장하고 구성합니다.

로그 저장소에 대한 일부 수정 사항은 다음과 같습니다.

- Elasticsearch 클러스터의 스토리지;
- shard가 전체 복제에서 복제 없이 클러스터의 데이터 노드에 복제되는 방법;
- Elasticsearch 데이터에 대한 외부 액세스를 허용합니다.



#### 참고

Elasticsearch 노드 축소는 지원되지 않습니다. 축소하면 Elasticsearch 포트가 실수로 삭제되어 shard가 할당되지 않고 복제 shard가 손실될 수 있습니다.

Elasticsearch는 메모리를 많이 사용하는 애플리케이션입니다. 클러스터 로깅 사용자 정의 리소스에서 달리 지정하지 않는 한 각 Elasticsearch 노드에는 메모리 요청 및 제한 모두에 16G의 메모리가 필요합니다. 초기 OpenShift Container Platform 노드 세트는 Elasticsearch 클러스터를 지원하기에 충분히 크지 않을 수 있습니다. 권장 메모리 이상으로 실행하려면 OpenShift Container Platform 클러스터에 노드를 추가해야 합니다.

각 Elasticsearch 노드는 더 낮은 메모리 설정으로 작동할 수 있지만 프로덕션 배포에는 권장되지 않습니다.

#### 3.3.1. 감사 로그를 로그 저장소로 전달

내부 OpenShift Container Platform Elasticsearch 로그 저장소는 감사 로그를 위한 보안 스토리지를 제공하지 않기 때문에 기본적으로 감사 로그는 내부 Elasticsearch 인스턴스에 저장되지 않습니다.

예를 들어 Kibana에서 감사 로그를 보기 위해 감사 로그를 내부 로그 저장소로 보내려면 Log Forward API를 사용해야 합니다.



## 중요

내부 OpenShift Container Platform Elasticsearch 로그 저장소는 감사 로그를 위한 보안 스토리지를 제공하지 않습니다. 감사 로그를 전달하는 시스템이 조직 및 정부 규정을 준수하고 올바르게 보호되도록 하는 것이 좋습니다. OpenShift Container Platform 클러스터 로깅은 이러한 규정을 준수하지 않습니다.

## 프로시저

Log Forward API를 사용하여 감사 로그를 내부 Elasticsearch 인스턴스로 전달하려면 다음을 수행합니다.

1. 로그 전달 CR YAML 파일을 생성하거나 기존 CR을 편집합니다.

- 모든 로그 유형을 내부 Elasticsearch 인스턴스로 보내는 CR을 생성합니다. 변경하지 않고 다음 예제를 사용할 수 있습니다.

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: instance
  namespace: openshift-logging
spec:
  pipelines: 1
  - name: all-to-default
    inputRefs:
      - infrastructure
      - application
      - audit
    outputRefs:
      - default
```

1. 파이프라인은 지정된 출력을 사용하여 전달할 로그 유형을 정의합니다. 기본 출력은 로그를 내부 Elasticsearch 인스턴스로 전달합니다.



## 참고

파이프라인에서 애플리케이션, 인프라 및 감사의 세 가지 유형의 로그를 모두 지정해야 합니다. 로그 유형을 지정하지 않으면 해당 로그가 저장되지 않고 손실됩니다.

- 기존 LogForwarding CR이 있는 경우 감사 로그의 기본 출력에 파이프 라인을 추가합니다. 기본 출력을 정의할 필요가 없습니다. 예를 들면 다음과 같습니다.

```
apiVersion: "logging.openshift.io/v1"
kind: ClusterLogForwarder
metadata:
  name: instance
  namespace: openshift-logging
spec:
  outputs:
    - name: elasticsearch-insecure
      type: "elasticsearch"
      url: elasticsearch-insecure.svc.messaging.cluster.local
      insecure: true
```

```

- name: elasticsearch-secure
  type: "elasticsearch"
  url: elasticsearch-secure.svc.messaging.cluster.local
  secret:
    name: es-audit
- name: secureforward-offcluster
  type: "forward"
  url: https://secureforward.offcluster.com:24224
  secret:
    name: secureforward
pipelines:
- name: container-logs
  inputRefs: application
  outputRefs:
    - secureforward-offcluster
- name: infra-logs
  inputRefs: infrastructure
  outputRefs:
    - elasticsearch-insecure
- name: audit-logs
  inputRefs: audit
  outputRefs:
    - elasticsearch-secure
    - default ①

```

① 이 파이프라인은 외부 인스턴스와 함께 내부 Elasticsearch 인스턴스로 감사 로그를 보냅니다.

## 추가 리소스

Log Forwarding API에 대한 자세한 내용은 [Log Forwarding API를 사용하여 로그 전달](#) 을 참조하십시오.

### 3.3.2. 로그 보존 시간 구성

인프라 로그, 애플리케이션 로그 및 감사 로그의 세 가지 로그 소스 각각에 대해 별도의 *보존 정책* 을 사용하여 기본 Elasticsearch 로그 저장소가 인덱스를 유지하는 기간을 지정할 수 있습니다. 클러스터 로깅 사용자 정의 리소스(CR)에서 **maxAge** 매개변수를 사용하여 구성하는 보존 정책은 Elasticsearch 롤오버 일정에 대해 고려되며 Elasticsearch가 롤오버된 인덱스를 삭제하는 시기를 결정합니다.

인덱스가 다음 조건 중 하나와 일치하면 Elasticsearch는 인덱스를 롤오버하여 현재 인덱스를 이동하고 새 인덱스를 만듭니다.

- 인덱스가 Elasticsearch CR의 **rollover.maxAge** 값보다 오래되었습니다.
- 인덱스 크기가 40GB × 기본 shard 수보다 큼니다.
- 인덱스 문서 수가 40960KB × 기본 shard 수보다 큼니다.

Elasticsearch 삭제 구성한 보존 정책에 따라 롤오버된 인덱스가 삭제됩니다.

로그 소스에 대한 보존 정책을 생성하지 않으면 기본적으로 7일 후에 로그가 삭제됩니다.





## 중요

세 개의 로그 소스 모두에 대해 보존 정책을 지정하지 않으면 보존 정책이 있는 소스의 로그만 저장됩니다. 예를 들어, 인프라 및 애플리케이션 로그에 대한 보존 정책을 설정했지만 감사 로그에 대해서는 보존 정책을 설정하지 않으면 감사 로그가 유지되지 않으며 Elasticsearch 또는 Kibana에 **감사** 인덱스가 없습니다.

## 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

## 프로시저

로그 보존 시간을 구성하려면 다음을 수행합니다.

- retentionPolicy** 매개변수를 추가하거나 수정하려면 클러스터 로깅 CR을 편집합니다.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
...
spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    retentionPolicy: 1
      application:
        maxAge: 1d
      infra:
        maxAge: 7d
      audit:
        maxAge: 7d
    elasticsearch:
      nodeCount: 3
  ...
```

- 1 Elasticsearch가 각 로그 소스를 유지해야 하는 시간을 지정합니다. 정수 및 시간 지정을 입력합니다(주(w), 시간(h/H), 분(m) 및 초(s)). 예를 들어 하루 동안은 **1d**입니다. **maxAge**보다 오래된 로그는 삭제됩니다. 기본적으로 로그는 7일 동안 유지됩니다.

- Elasticsearch 사용자 정의 리소스(CR)에서 설정을 확인할 수 있습니다.

예를 들어, 클러스터 로깅 Operator는 8시간마다 인프라 로그의 활성 인덱스를 롤오버하는 설정을 포함하는 보존 정책을 구성하기 위해 다음 Elasticsearch CR을 업데이트했으며 롤오버된 인덱스는 롤오버 후 7일 후에 삭제됩니다. OpenShift Container Platform은 15분마다 인덱스를 롤오버해야 하는지 확인합니다.

```
apiVersion: "logging.openshift.io/v1"
kind: "Elasticsearch"
metadata:
  name: "elasticsearch"
spec:
  ...
  indexManagement:
    policies: 1
      - name: infra-policy
        phases:
```

```

delete:
  minAge: 7d 2
hot:
  actions:
    rollover:
      maxAge: 8h 3
pollInterval: 15m 4
...

```

- 1 각 로그 소스에 대해 보존 정책은 해당 소스에 대한 로그를 삭제하고 롤오버할 시기를 나타냅니다.
- 2 OpenShift Container Platform이 롤오버된 인덱스를 삭제하는 경우 이 설정은 클러스터 로깅 CR에서 설정한 **maxAge**입니다.
- 3 인덱스를 롤오버할 때 고려해야 할 OpenShift Container Platform의 인덱스 수명입니다. 이 값은 클러스터 로깅 CR에서 설정한 **maxAge**에서 결정됩니다.
- 4 OpenShift Container Platform에서 인덱스를 롤오버해야 하는지 확인하는 경우 이 설정은 기본값이며 변경할 수 없습니다.



#### 참고

Elasticsearch CR 수정은 지원되지 않습니다. 보존 정책에 대한 모든 변경은 클러스터 로깅 CR에서 수행해야 합니다.

Elasticsearch Operator는 CronJob을 배포하여 **pollInterval**을 사용하여 예약된 정의된 정책을 사용하여 각 매핑에 대한 인덱스를 롤오버합니다.

```
$ oc get cronjobs
```

#### 출력 예

NAME	SCHEDULE	SUSPEND	ACTIVE	LAST SCHEDULE	AGE
elasticsearch-delete-app	*/15 * * * *	False	0	<none>	27s
elasticsearch-delete-audit	*/15 * * * *	False	0	<none>	27s
elasticsearch-delete-infra	*/15 * * * *	False	0	<none>	27s
elasticsearch-rollover-app	*/15 * * * *	False	0	<none>	27s
elasticsearch-rollover-audit	*/15 * * * *	False	0	<none>	27s
elasticsearch-rollover-infra	*/15 * * * *	False	0	<none>	27s

### 3.3.3. 로그 저장소에 대한 CPU 및 메모리 요청 구성

각 구성 요소 사양을 통해 CPU 및 메모리 요청을 조정할 수 있습니다. Elasticsearch Operator가 환경에 충분한 값을 설정하므로 이러한 값을 수동으로 조정할 필요는 없습니다.

각 Elasticsearch 노드는 더 낮은 메모리 설정으로 작동할 수 있지만 프로덕션 배포에는 권장되지 않습니다. 프로덕션 용도의 경우 각 포드에 할당된 기본 16Gi 이상이어야 합니다. 가급적 포트당 최대 64Gi를 할당해야 합니다.

#### 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

## 프로시저

1. **openshift-logging** 프로젝트에서 클러스터 로깅 사용자 정의 리소스(CR)를 편집합니다.

```
$ oc edit ClusterLogging instance

apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
....
spec:
  logStore:
    type: "elasticsearch"
    elasticsearch:
      resources: ❶
        limits:
          memory: 16Gi
        requests:
          cpu: 500m
          memory: 16Gi
```

- ❶ 필요에 따라 CPU 및 메모리 요청을 지정합니다. 이 값을 비워두면 Elasticsearch Operator는 대부분의 배포에 충분한 기본값을 설정합니다.

Elasticsearch 메모리 양을 조정하는 경우 요청 값과 제한 값을 모두 변경해야 합니다.

예를 들면 다음과 같습니다.

```
resources:
  limits:
    memory: "32Gi"
  requests:
    cpu: "8"
    memory: "32Gi"
```

Kubernetes는 일반적으로 노드 구성을 준수하며 Elasticsearch가 지정된 제한을 사용하는 것을 허용하지 않습니다. 요청 및 제한에 대해 동일한 값을 설정하면 노드에 사용 가능한 메모리가 있다고 가정하고 Elasticsearch가 원하는 메모리를 사용할 수 있습니다.

### 3.3.4. 로그 저장소에 대한 복제 정책 구성

Elasticsearch shard가 클러스터의 데이터 노드에 복제되는 방법을 정의할 수 있습니다.

## 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

## 프로시저

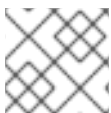
1. **openshift-logging** 프로젝트에서 클러스터 로깅 사용자 정의 리소스(CR)를 편집합니다.

```
$ oc edit clusterlogging instance
```

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
....
spec:
  logStore:
    type: "elasticsearch"
    elasticsearch:
      redundancyPolicy: "SingleRedundancy" 1
```

**1** shard에 대한 중복 정책을 지정합니다. 변경 사항을 저장하면 변경 사항이 적용됩니다.

- **FullRedundancy.** Elasticsearch는 각 인덱스의 기본 shard를 모든 데이터 노드에 완전히 복제합니다. 이는 가장 높은 안전성을 제공하지만 필요한 디스크 양이 가장 많고 성능이 가장 낮습니다.
- **MultipleRedundancy.** Elasticsearch는 각 인덱스의 기본 shard를 데이터 노드의 절반으로 완전히 복제합니다. 이는 안전성과 성능 사이의 좋은 균형을 제공합니다.
- **SingleRedundancy.** Elasticsearch는 각 인덱스에 대해 기본 shard의 사본 하나를 만듭니다. 두 개 이상의 데이터 노드가 존재하는 한 항상 로그를 사용할 수 있고 복구할 수 있습니다. 5개 이상의 노드를 사용하는 경우 MultipleRedundancy보다 성능이 향상됩니다. 단일 Elasticsearch 노드 배포에는 이 정책을 적용할 수 없습니다.
- **ZeroRedundancy.** Elasticsearch는 기본 shard의 사본을 만들지 않습니다. 노드가 다운되거나 실패한 경우 로그를 사용할 수 없거나 손실될 수 있습니다. 안전보다 성능이 더 중요하거나 자체 디스크/PVC 백업/복원 전략을 구현한 경우 이 모드를 사용합니다.



## 참고

인덱스 템플릿의 기본 shard 수는 Elasticsearch 데이터 노드 수와 같습니다.

### 3.3.5. 로그 저장소에 대한 영구 스토리지 구성

Elasticsearch에는 영구 스토리지가 필요합니다. 스토리지가 빠를수록 Elasticsearch 성능이 빨라집니다.



## 주의

Lucene은 NFS가 제공하지 않는 파일 시스템 동작을 사용하므로 Elasticsearch 스토리지에서는 NFS 스토리지를 볼륨 또는 영구 볼륨(또는 Gluster와 같은 NAS를 통해)으로 사용할 수 없습니다. 데이터 손상 및 기타 문제가 발생할 수 있습니다.

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

#### 프로시저

1. 클러스터 로깅 CR을 편집하여 클러스터의 각 데이터 노드가 영구 볼륨 클레임에 바인딩되도록 지정합니다.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
....

spec:
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      storage:
        storageClassName: "gp2"
        size: "200G"
```

이 예는 클러스터의 각 데이터 노드가 AWS General Purpose SSD(gp2) 스토리지의 "200G"를 요청하는 영구 볼륨 클레임에 바인딩되도록 지정합니다.

### 3.3.6. emptyDir 스토리지에 대한 로그 저장소 구성

emptyDir을 로그 저장소와 함께 사용하면 재시작 시 포드의 모든 데이터가 손실되는 임시 배포가 생성됩니다.



#### 참고

emptyDir를 사용할 때 로그 스토리지가 다시 시작되거나 재배포되면 데이터가 손실됩니다.

#### 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

#### 프로시저

1. emptyDir을 지정하려면 클러스터 로깅 CR을 편집합니다.

```
spec:
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      storage: {}
```

### 3.3.7. Elasticsearch 롤링 클러스터 재시작 수행

**elasticsearch** 구성맵 또는 **elasticsearch-\*** 배치 구성을 변경할 때 롤링 재시작을 수행합니다.

또한 Elasticsearch 포드가 실행되는 노드를 재부팅해야 하는 경우 롤링 재시작이 권장됩니다.

### 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

### 프로시저

롤링 클러스터 재시작을 수행하려면 다음을 수행합니다.

1. **openshift-logging** 프로젝트로 변경합니다.

```
$ oc project openshift-logging
```

2. 다음 명령을 사용하여 Elasticsearch에서 CA 인증서를 추출하고 **admin-ca** 파일에 작성합니다.

```
$ oc extract secret/elasticsearch --to=. --keys=admin-ca
```

```
admin-ca
```

3. shard 동기화 플러시를 수행하여 종료하기 전에 디스크에 쓰기 대기 중인 작업이 없는지 확인하십시오.

```
$ oc exec <any_es_pod_in_the_cluster> -c elasticsearch -- curl -s --cacert
/etc/elasticsearch/secret/admin-ca --cert /etc/elasticsearch/secret/admin-cert --key
/etc/elasticsearch/secret/admin-key -XPOST 'https://localhost:9200/_flush/synced'
```

예를 들면 다음과 같습니다.

```
$ oc exec -c elasticsearch-cdm-5ceex6ts-1-dcd6c4c7c-jpw6 -- curl -s --cacert
/etc/elasticsearch/secret/admin-ca --cert /etc/elasticsearch/secret/admin-cert --key
/etc/elasticsearch/secret/admin-key -XPOST 'https://localhost:9200/_flush/synced'
```

4. OpenShift Container Platform **es\_util** 도구를 사용하여 의도적으로 노드를 중단할 때 shard 밸런싱을 방지합니다.

```
$ oc exec <any_es_pod_in_the_cluster> -c elasticsearch -- es_util --query=_cluster/settings -
XPUT 'https://localhost:9200/_cluster/settings' -d '{ "transient": {
"cluster.routing.allocation.enable" : "none" } }'
```

예를 들면 다음과 같습니다.

```
$ oc exec elasticsearch-cdm-5ceex6ts-1-dcd6c4c7c-jpw6 -c elasticsearch -- es_util --
query=_cluster/settings?pretty=true -XPUT 'https://localhost:9200/_cluster/settings' -d '{
"transient": { "cluster.routing.allocation.enable" : "none" } }'
```

### 출력 예

```
{
  "acknowledged" : true,
  "persistent" : { },
```

```

    "transient" : {
      "cluster" : {
        "routing" : {
          "allocation" : {
            "enable" : "none"
          }
        }
      }
    }
  }
}
}
}

```

5. 완료되면 ES 클러스터에 대해 보유한 각 배포에 대해 다음을 수행합니다.

- a. 기본적으로 OpenShift Container Platform Elasticsearch 클러스터는 노드로의 롤아웃을 차단합니다. 다음 명령을 사용하여 롤아웃을 허용하고 포트가 변경 사항을 선택하도록 합니다.

```
$ oc rollout resume deployment/<deployment-name>
```

예를 들면 다음과 같습니다.

```
$ oc rollout resume deployment/elasticsearch-cdm-0-1
```

```
deployment.extensions/elasticsearch-cdm-0-1 resumed
```

새 포트가 배포되었습니다. 포트에 컨테이너가 준비되면 다음 배포로 이동할 수 있습니다.

```
$ oc get pods | grep elasticsearch-*
```

**출력 예**

NAME	READY	STATUS	RESTARTS	AGE
elasticsearch-cdm-5ceex6ts-1-dcd6c4c7c-jpw6k	2/2	Running	0	22h
elasticsearch-cdm-5ceex6ts-2-f799564cb-l9mj7	2/2	Running	0	22h
elasticsearch-cdm-5ceex6ts-3-585968dc68-k7kjr	2/2	Running	0	22h

- b. 완료되면 롤아웃을 허용하지 않도록 포드를 재설정합니다.

```
$ oc rollout pause deployment/<deployment-name>
```

예를 들면 다음과 같습니다.

```
$ oc rollout pause deployment/elasticsearch-cdm-0-1
```

```
deployment.extensions/elasticsearch-cdm-0-1 paused
```

- c. Elasticsearch 클러스터가 녹색 상태인지 확인하십시오.

```
$ oc exec <any_es_pod_in_the_cluster> -c elasticsearch -- es_util --
query=_cluster/health?pretty=true
```



## 참고

이전 명령에서 사용한 Elasticsearch 포트에서 롤아웃을 수행한 경우 해당 포트가 더 이상 존재하지 않으며 여기에 새 포트 이름이 필요합니다.

예를 들면 다음과 같습니다.

```
$ oc exec elasticsearch-cdm-5ceex6ts-1-dcd6c4c7c-jpw6 -c elasticsearch -- es_util --
query=_cluster/health?pretty=true
```

## 출력 예

```
{
  "cluster_name" : "elasticsearch",
  "status" : "green", ①
  "timed_out" : false,
  "number_of_nodes" : 3,
  "number_of_data_nodes" : 3,
  "active_primary_shards" : 8,
  "active_shards" : 16,
  "relocating_shards" : 0,
  "initializing_shards" : 0,
  "unassigned_shards" : 1,
  "delayed_unassigned_shards" : 0,
  "number_of_pending_tasks" : 0,
  "number_of_in_flight_fetch" : 0,
  "task_max_waiting_in_queue_millis" : 0,
  "active_shards_percent_as_number" : 100.0
}
```

① 계속하기 전에 이 매개 변수가 녹색 인지 확인하십시오.

6. Elasticsearch 구성 맵을 변경한 경우 각 Elasticsearch 포트에 대해 이 단계를 반복합니다.

7. 클러스터에 대한 모든 배포가 롤아웃되면 shard 밸런싱을 다시 활성화합니다.

```
$ oc exec <any_es_pod_in_the_cluster> -c elasticsearch -- es_util --query=_cluster/settings -
XPUT 'https://localhost:9200/_cluster/settings' -d '{ "transient": {
"cluster.routing.allocation.enable" : "none" } }'
```

예를 들면 다음과 같습니다.

```
$ oc exec elasticsearch-cdm-5ceex6ts-1-dcd6c4c7c-jpw6 -c elasticsearch -- es_util --
query=_cluster/settings?pretty=true -XPUT 'https://localhost:9200/_cluster/settings' -d '{
"transient": { "cluster.routing.allocation.enable" : "all" } }'
```

## 출력 예

```
{
  "acknowledged" : true,
  "persistent" : { },
  "transient" : {
```



```

    "cluster" : {
      "routing" : {
        "allocation" : {
          "enable" : "all"
        }
      }
    }
  }
}
}
}

```

### 3.3.8. 로그 저장소 서비스를 경로로 노출

기본적으로 클러스터 로깅과 함께 배포된 로그 저장소는 로깅 클러스터 외부에서 액세스할 수 없습니다. 데이터에 액세스하는 도구의 로그 저장소 서비스에 대한 외부 액세스를 위해 재암호화 종료로 경로를 활성화할 수 있습니다.

외부에서는 재암호화 경로, OpenShift Container Platform 토큰 및 설치된 로그 저장소 CA 인증서를 만들어 로그 저장소에 액세스할 수 있습니다. 그런 후 다음을 포함하는 cURL 요청으로 로그 저장소 서비스를 호스팅하는 노드에 액세스합니다.

- 승인: **Bearer \${token}**
- Elasticsearch 재암호화 경로 및 [Elasticsearch API 요청](#).

내부적으로 다음 명령 중 하나를 사용하여 얻을 수 있는 로그 저장소 클러스터 IP를 사용하여 로그 저장소 서비스에 액세스할 수 있습니다.

```
$ oc get service elasticsearch -o jsonpath={.spec.clusterIP} -n openshift-logging
```

출력 예

```
172.30.183.229
```

```
$ oc get service elasticsearch -n openshift-logging
```

출력 예

```

NAME          TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
elasticsearch ClusterIP   172.30.183.229 <none>       9200/TCP   22h

```

다음과 유사한 명령을 사용하여 클러스터 IP 주소를 확인할 수 있습니다.

```
$ oc exec elasticsearch-cdm-oplnhinv-1-5746475887-fj2f8 -n openshift-logging -- curl -tlsv1.2 --insecure -H "Authorization: Bearer ${token}" "https://172.30.183.229:9200/_cat/health"
```

출력 예

```

% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100  29 100  29  0  0  108  0 --:--:-- --:--:-- --:--:-- 108

```

사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.
- 로그에 액세스하려면 프로젝트에 액세스할 수 있어야 합니다.

## 프로시저

로그 저장소를 외부에 노출하려면 다음을 수행합니다.

1. **openshift-loggin** 프로젝트로 변경합니다.

```
$ oc project openshift-logging
```

2. 로그 저장소에서 CA 인증서를 추출하고 **admin-ca** 파일에 씁니다.

```
$ oc extract secret/elasticsearch --to=. --keys=admin-ca
```

### 출력 예

```
admin-ca
```

3. 로그 저장소 서비스의 경로를 YAML 파일로 생성합니다.

- a. 다음을 사용하여 YAML 파일을 생성합니다.

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: elasticsearch
  namespace: openshift-logging
spec:
  host:
  to:
    kind: Service
    name: elasticsearch
  tls:
    termination: reencrypt
    destinationCACertificate: | 1
```

- 1 로그 저장소 CA 인증서를 추가하거나 다음 단계에서 명령을 사용합니다. 일부 재암호화 경로에 필요한 **spec.tls.key**, **spec.tls.certificate** 및 **spec.tls.caCertificate** 매개 변수를 설정할 필요는 없습니다.

- b. 다음 명령을 실행하여 작성한 경로 YAML에 로그 저장소 CA 인증서를 추가합니다.

```
$ cat ./admin-ca | sed -e "s/^ /" >> <file-name>.yaml
```

- c. 경로를 만듭니다.

```
$ oc create -f <file-name>.yaml
```

### 출력 예

```
route.route.openshift.io/elasticsearch created
```

## 4. Elasticsearch 서비스가 노출되어 있는지 확인하십시오.

- a. 요청에 사용될 이 ServiceAccount의 토큰을 가져옵니다.

```
$ token=$(oc whoami -t)
```

- b. 생성한
- elasticsearch**
- 경로를 환경 변수로 설정합니다.

```
$ routeES=$(oc get route elasticsearch -o jsonpath={.spec.host})
```

- c. 경로가 성공적으로 생성되었는지 확인하려면 노출된 경로를 통해 Elasticsearch에 액세스하는 다음 명령을 실행합니다.

```
$ curl -tlsv1.2 --insecure -H "Authorization: Bearer ${token}"
"https://${routeES}/.operations.*/_search?size=1" | jq
```

응답은 다음과 유사하게 나타납니다.

## 출력 예

```
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
100  944  100  944    0    0   62    0 0:00:15 0:00:15 --:--:-- 204
{
  "took": 441,
  "timed_out": false,
  "_shards": {
    "total": 3,
    "successful": 3,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 89157,
    "max_score": 1,
    "hits": [
      {
        "_index": ".operations.2019.03.15",
        "_type": "com.example.viaq.common",
        "_id": "ODdiNWlyYzAtMjg5Ni0TAAtNWE3MDY1MjMzNTc3",
        "_score": 1,
        "_source": {
          "_SOURCE_MONOTONIC_TIMESTAMP": "673396",
          "systemd": {
            "t": {
              "BOOT_ID": "246c34ee9cdeecb41a608e94",
              "MACHINE_ID": "e904a0bb5efd3e36badee0c",
              "TRANSPORT": "kernel"
            },
            "u": {
              "SYSLOG_FACILITY": "0",
              "SYSLOG_IDENTIFIER": "kernel"
            }
          }
        },
        "level": "info",
```

```

    "message": "acpiphp: Slot [30] registered",
    "hostname": "localhost.localdomain",
    "pipeline_metadata": {
      "collector": {
        "ipaddr4": "10.128.2.12",
        "ipaddr6": "fe80::xx:xxx:fe4c:5b09",
        "inputname": "fluent-plugin-systemd",
        "name": "fluentd",
        "received_at": "2019-03-15T20:25:06.273017+00:00",
        "version": "1.3.2 1.6.0"
      }
    },
    "@timestamp": "2019-03-15T20:00:13.808226+00:00",
    "viaq_msg_id": "ODdiNWlyYzAtMYTAhNWE3MDY1MjMzNTc3"
  }
}
]
}
}

```

### 3.4. 로그 시각화 프로그램 구성

OpenShift Container Platform은 Kibana를 사용하여 클러스터 로깅으로 수집된 로그 데이터를 표시합니다.

중복성을 위해 Kibana를 확장하고 Kibana 노드의 CPU 및 메모리를 구성할 수 있습니다.

#### 3.4.1. CPU 및 메모리 제한 구성

클러스터 로깅 구성 요소를 사용하면 CPU 및 메모리 제한을 모두 조정할 수 있습니다.

##### 프로시저

1. **openshift-logging** 프로젝트에서 클러스터 로깅 사용자 정의 리소스(CR)를 편집합니다.

```
$ oc edit ClusterLogging instance -n openshift-logging
```

```

apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
....

spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 2
      resources: ①
        limits:
          memory: 2Gi
        requests:

```

```

    cpu: 200m
    memory: 2Gi
  storage:
    storageClassName: "gp2"
    size: "200G"
  redundancyPolicy: "SingleRedundancy"
visualization:
  type: "kibana"
  kibana:
    resources: ❷
    limits:
      memory: 1Gi
    requests:
      cpu: 500m
      memory: 1Gi
  proxy:
    resources: ❸
    limits:
      memory: 100Mi
    requests:
      cpu: 100m
      memory: 100Mi
  replicas: 2
curation:
  type: "curator"
  curator:
    resources: ❹
    limits:
      memory: 200Mi
    requests:
      cpu: 200m
      memory: 200Mi
  schedule: "*/10 * * * *"
collection:
  logs:
    type: "fluentd"
    fluentd:
      resources: ❺
      limits:
        memory: 736Mi
      requests:
        cpu: 200m
        memory: 736Mi

```

- ❶ 필요에 따라 로그 저장소에 대한 CPU 및 메모리 제한 및 요청을 지정합니다. Elasticsearch의 경우 요청 값과 제한 값을 모두 조정해야 합니다.
- ❷ ❸ 필요에 따라 로그 시각화 프로그램에 대한 CPU 및 메모리 제한 및 요청을 지정합니다.
- ❹ 필요에 따라 로그 큐레이터에 대한 CPU 및 메모리 제한 및 요청을 지정합니다.
- ❺ 필요에 따라 로그 수집기에 대한 CPU 및 메모리 제한 및 요청을 지정합니다.

### 3.4.2. 로그 시각화 프로그램 노드의 확장성 중복

중복성에 대해 로그 시각화 프로그램을 호스팅하는 포드를 확장할 수 있습니다.

## 프로시저

1. **openshift-logging** 프로젝트에서 클러스터 로깅 사용자 정의 리소스(CR)를 편집합니다.

```
$ oc edit ClusterLogging instance

$ oc edit ClusterLogging instance

apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
....

spec:
  visualization:
    type: "kibana"
    kibana:
      replicas: 1 1
```

- 1 Kibana 노드의 수를 지정합니다.

## 3.5. 클러스터 로깅 스토리지 구성

Elasticsearch는 메모리를 많이 사용하는 애플리케이션입니다. 기본 클러스터 로깅 설치 시 메모리 요청 및 메모리 제한 모두에 16G 메모리를 배포합니다. 초기 OpenShift Container Platform 노드 세트는 Elasticsearch 클러스터를 지원하기에 충분히 크지 않을 수 있습니다. 권장 메모리 이상으로 실행하려면 OpenShift Container Platform 클러스터에 노드를 추가해야 합니다. 각 Elasticsearch 노드는 더 낮은 메모리 설정으로 작동할 수 있지만 프로덕션 배포에는 권장되지 않습니다.

### 3.5.1. 클러스터 로깅 및 OpenShift Container Platform에 대한 스토리지 고려 사항

각 Elasticsearch 배포에는 데이터 노드당 하나의 데이터 볼륨이 있어야 영구 볼륨이 필요합니다. OpenShift 컨테이너 플랫폼에서는 영구 볼륨 클레임을 사용합니다.

Elasticsearch Operator는 Elasticsearch 리소스 이름을 사용하여 PVC의 이름을 지정합니다. 자세한 내용은 영구 Elasticsearch 스토리지를 참조하십시오.

Fluentd는 **systemd journal** 및 **/var/log/containers/**의 모든 로그를 Elasticsearch에 제공합니다.

따라서 미리 필요한 데이터 양과 애플리케이션 로그 데이터를 집계하는 것을 고려하십시오. 일부 Elasticsearch 사용자는 절대 스토리지 소비량을 항상 약 50%에서 70% 미만으로 유지해야 한다는 것을 알게 되었습니다. 이는 대규모 병합 작업 중에 Elasticsearch가 응답하지 않는 것을 방지합니다.

기본적으로 85%에서 Elasticsearch는 새 데이터를 노드에 할당하는 것을 중지하고 90%에서 Elasticsearch는 기존 shard를 가능한 경우 해당 노드에서 다른 노드로 재배포하려고 시도합니다. 그러나 사용 가능한 용량이 85% 미만인 노드가 없는 경우 Elasticsearch는 새 인덱스 생성을 효과적으로 거부하고 RED가 됩니다.



## 참고

이 낮은 워터마크 값과 높은 워터마크 값은 현재 릴리스에서 Elasticsearch 기본값입니다. 이 값을 수정할 수 있지만 경고에도 수정 사항을 적용해야 합니다. 경고는 이러한 기본값을 기반으로 합니다.

### 3.6. 클러스터 로깅 구성 요소에 대한 CPU 및 메모리 제한 구성

필요에 따라 각 클러스터 로깅 구성 요소에 대한 CPU 및 메모리 제한을 모두 구성할 수 있습니다.

#### 3.6.1. CPU 및 메모리 제한 구성

클러스터 로깅 구성 요소를 사용하면 CPU 및 메모리 제한을 모두 조정할 수 있습니다.

#### 프로시저

1. **openshift-logging** 프로젝트에서 클러스터 로깅 사용자 정의 리소스(CR)를 편집합니다.

```
$ oc edit ClusterLogging instance -n openshift-logging
```

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
....
spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 2
      resources: 1
      limits:
        memory: 2Gi
      requests:
        cpu: 200m
        memory: 2Gi
    storage:
      storageClassName: "gp2"
      size: "200G"
      redundancyPolicy: "SingleRedundancy"
  visualization:
    type: "kibana"
    kibana:
      resources: 2
      limits:
        memory: 1Gi
      requests:
        cpu: 500m
        memory: 1Gi
    proxy:
      resources: 3
```

```

limits:
  memory: 100Mi
requests:
  cpu: 100m
  memory: 100Mi
replicas: 2
curation:
  type: "curator"
  curator:
    resources: ④
    limits:
      memory: 200Mi
    requests:
      cpu: 200m
      memory: 200Mi
    schedule: "*/10 * * * *"
collection:
  logs:
    type: "fluentd"
    fluentd:
      resources: ⑤
      limits:
        memory: 736Mi
      requests:
        cpu: 200m
        memory: 736Mi

```

- ① 필요에 따라 로그 저장소에 대한 CPU 및 메모리 제한 및 요청을 지정합니다. Elasticsearch의 경우 요청 값과 제한 값을 모두 조정해야 합니다.
- ② ③ 필요에 따라 로그 시각화 프로그램에 대한 CPU 및 메모리 제한 및 요청을 지정합니다.
- ④ 필요에 따라 로그 큐레이터에 대한 CPU 및 메모리 제한 및 요청을 지정합니다.
- ⑤ 필요에 따라 로그 수집기에 대한 CPU 및 메모리 제한 및 요청을 지정합니다.

### 3.7. 공차를 사용하여 클러스터 로깅 포드 배치 제어

오류 및 허용 오차를 사용하여 클러스터 로깅 포드가 특정 노드에서 실행되고 해당 노드에서 다른 워크로드를 실행할 수 없도록 할 수 있습니다.

오류 및 허용 오차는 간단한 **키:값** 쌍입니다. 노드의 오류는 노드가 오류를 허용하지 않는 모든 포드를 거절하도록 지시합니다.

키는 최대 253자의 문자열이며 값은 최대 63자의 문자열입니다. 문자열은 문자 또는 숫자로 시작해야 하며 문자, 숫자, 하이픈, 점 및 밑줄을 포함할 수 있습니다.

#### 허용 오차가 있는 샘플 클러스터 로깅 CR

```

apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: openshift-logging

```



```

spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 1
      tolerations: ①
      - key: "logging"
        operator: "Exists"
        effect: "NoExecute"
        tolerationSeconds: 6000
    resources:
      limits:
        memory: 8Gi
      requests:
        cpu: 100m
        memory: 1Gi
      storage: {}
      redundancyPolicy: "ZeroRedundancy"
  visualization:
    type: "kibana"
    kibana:
      tolerations: ②
      - key: "logging"
        operator: "Exists"
        effect: "NoExecute"
        tolerationSeconds: 6000
    resources:
      limits:
        memory: 2Gi
      requests:
        cpu: 100m
        memory: 1Gi
      replicas: 1
  collection:
    logs:
      type: "fluentd"
      fluentd:
        tolerations: ③
        - key: "logging"
          operator: "Exists"
          effect: "NoExecute"
          tolerationSeconds: 6000
    resources:
      limits:
        memory: 2Gi
      requests:
        cpu: 100m
        memory: 1Gi

```

- ① 이 허용 오차는 Elasticsearch 포트에 추가됩니다.
- ② 이 허용 오차는 Kibana 포트에 추가됩니다.
- ③ 이 허용 오차는 로깅 수집기 포트에 추가됩니다.

### 3.7.1. 허용 오차를 사용하여 로그 저장소 포드 배치 제어

포드의 허용 오차를 사용하여 로그 저장소 포드가 실행되는 노드를 제어하고 다른 워크로드가 해당 노드를 사용하지 못하게 할 수 있습니다.

클러스터 로깅 사용자 정의 리소스(CR)를 통해 로그 저장소 포드에 허용 오차를 적용하고 노드 스펙을 통해 노드에 오류를 적용합니다. 노드의 오류는 노드가 오류를 허용하지 않는 모든 포드를 거절하도록 지시하는 **키:값** 쌍입니다. 다른 포드에 없는 특정 **키:값** 쌍을 사용하면 해당 노드에서 로그 저장소 포드만 실행할 수 있습니다.

기본적으로 로그 저장소 포드에는 다음과 같은 허용 오차가 있습니다.

```
tolerations:
- effect: "NoExecute"
  key: "node.kubernetes.io/disk-pressure"
  operator: "Exists"
```

#### 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

#### 프로시저

- 다음 명령을 사용하여 클러스터 로깅 포드를 예약하려는 노드에 오류를 추가합니다.

```
$ oc adm taint nodes <node-name> <key>=<value>:<effect>
```

예를 들면 다음과 같습니다.

```
$ oc adm taint nodes node1 elasticsearch=node:NoExecute
```

이 예제는 키 **elasticsearch**, 값 **node** 및 오류 효과 **NoExecute**가 있는 **node1**에 오류를 배치합니다. **NoExecute** 효과가 있는 노드는 오류와 일치하는 포드만 예약하고 일치하지 않는 기존 포드는 제거합니다.

- Elasticsearch Pod에 대한 허용 오차를 구성하려면 클러스터 로깅 사용자 정의 리소스(CR)의 **로그 저장소** 섹션을 편집합니다.

```
logStore:
  type: "elasticsearch"
  elasticsearch:
    nodeCount: 1
    tolerations:
      - key: "elasticsearch" ①
        operator: "Exists" ②
        effect: "NoExecute" ③
        tolerationSeconds: 6000 ④
```

- ① 노드에 추가한 키를 지정합니다.
- ② 노드에 키 **elasticsearch**가 있는 오류가 필요하도록 **Exists** Operator를 지정합니다.
- ③ **NoExecute** 효과를 지정합니다.

- 4 선택적으로 **tolerationSeconds** 매개변수를 지정하여 Pod가 제거되기 전에 포드가 노드에 바인드될 수 있는 시간을 설정합니다.

이 허용 오차는 **oc adm taint** 명령으로 생성된 오류와 일치합니다. 이 허용 오차가 있는 포드를 **node1**에 예약할 수 있습니다.

### 3.7.2. 허용 오차를 사용하여 로그 시각화 프로그램 포드 배치 제어

포드의 허용 오차를 사용하여 로그 시각화 프로그램 포드가 실행되는 노드를 제어하고 다른 워크로드에서 해당 노드를 사용하지 못하게 할 수 있습니다.

클러스터 로깅 사용자 정의 리소스(CR)를 통해 로그 시각화 프로그램 포드에 허용 오차를 적용하고 노드 사양을 통해 노드에 오류를 적용합니다. 노드의 오류는 노드가 오류를 허용하지 않는 모든 포드를 거절하도록 지시하는 **키:값 쌍**입니다. 다른 포드에 없는 특정 **키:값 쌍**을 사용하면 해당 노드에서 Kibana 포드만 실행할 수 있습니다.

#### 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

#### 프로시저

- 다음 명령을 사용하여 로그 시각화 프로그램 포드를 예약하려는 노드에 오류를 추가합니다.

```
$ oc adm taint nodes <node-name> <key>=<value>:<effect>
```

예를 들면 다음과 같습니다.

```
$ oc adm taint nodes node1 kibana=node:NoExecute
```

이 예제는 키 **kibana**, 값 **node** 및 오류 효과 **NoExecute**가 있는 **node1**에 오류를 배치합니다. **NoExecute** 오류 효과를 사용해야 합니다. **NoExecute**는 오류와 일치하는 포드만 예약하고 일치하지 않는 기존 포드는 제거합니다.

- Kibana Pod에 대한 허용 오차를 구성하려면 클러스터 로깅 사용자 정의 리소스(CR)의 **시각화** 섹션을 편집합니다.

```
visualization:
  type: "kibana"
  kibana:
    tolerations:
      - key: "kibana" 1
        operator: "Exists" 2
        effect: "NoExecute" 3
        tolerationSeconds: 6000 4
```

- 1 노드에 추가한 키를 지정합니다.
- 2 키/값/효과 매개변수가 일치하도록 **Exists** Operator를 지정합니다.
- 3 **NoExecute** 효과를 지정합니다.
- 4 선택적으로 **tolerationSeconds** 매개변수를 지정하여 Pod가 제거되기 전에 포드가 노드에 바인드될 수 있는 시간을 설정합니다.

이 허용 오차는 **oc adm taint** 명령으로 생성된 오류와 일치합니다. 이 허용 오차가 있는 포드는 **node1**에 예약할 수 있습니다.

### 3.7.3. 허용 오차를 사용하여 로깅 수집기 포드 배치 제어

포드의 허용 오차를 사용하여 로깅 수집기 포드가 실행되는 노드를 확인하고 다른 워크로드가 해당 노드를 사용하지 못하게 할 수 있습니다.

클러스터 로깅 사용자 정의 리소스(CR)를 통해 로깅 수집기 포드에 허용 오차를 적용하고 노드 사양을 통해 노드에 오류를 적용합니다. 메모리 및 CPU 문제와 같은 항목에 대해 포드가 제거되지 않도록 오류 및 허용 오차를 사용할 수 있습니다.

기본적으로 로깅 수집기 포드에는 다음과 같은 허용 오차가 있습니다.

```
tolerations:
- key: "node-role.kubernetes.io/master"
  operator: "Exists"
  effect: "NoExecute"
```

#### 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

#### 프로시저

- 다음 명령을 사용하여 로깅 수집기 포드가 로깅 수집기 포드를 예약할 노드에 오류를 추가합니다.

```
$ oc adm taint nodes <node-name> <key>=<value>:<effect>
```

예를 들면 다음과 같습니다.

```
$ oc adm taint nodes node1 collector=node:NoExecute
```

이 예에서는 키 수집기, 값 **node** 및 오류 효과 **NoExecute**가 있는 **node1**에 오류를 배치합니다. **NoExecute** 오류 효과를 사용해야 합니다. **NoExecute**는 오류와 일치하는 포드만 예약하고 일치하지 않는 기존 포드는 제거합니다.

- 클러스터 로깅 사용자 정의 리소스(CR)의 **컬렉션** 섹션을 편집하여 로깅 수집기 포드에 대한 허용을 구성합니다.

```
collection:
  logs:
    type: "fluentd"
  rsyslog:
    tolerations:
      - key: "collector" 1
        operator: "Exists" 2
        effect: "NoExecute" 3
        tolerationSeconds: 6000 4
```

- 1 노드에 추가한 키를 지정합니다.

- 2 키/값/효과 매개변수가 일치하도록 **Exists** Operator를 지정합니다.
- 3 **NoExecute** 효과를 지정합니다.
- 4 선택적으로 **tolerationSeconds** 매개변수를 지정하여 Pod가 제거되기 전에 포드가 노드에 바인드될 수 있는 시간을 설정합니다.

이 허용 오차는 **oc adm taint** 명령으로 생성된 오류와 일치합니다. 이 허용 오차가 있는 포드는 **node1**에 예약할 수 있습니다.

### 3.7.4. 추가 리소스

오류 및 허용 오차에 대한 자세한 내용은 [노드 오류를 사용하여 포드 배치 제어](#)를 참조하십시오.

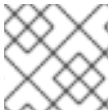
## 3.8. 노드 선택기로 클러스터 로깅 리소스 이동

노드 선택기를 사용하여 Elasticsearch, Kibana 및 Curator 포드를 다른 노드에 배포합니다.

### 3.8.1. 클러스터 로깅 리소스 이동

클러스터 로깅 구성 요소, Elasticsearch, Kibana 및 Curator의 일부 또는 전체에 대한 포드를 다른 노드에 배포하도록 클러스터 로깅 Operator를 구성할 수 있습니다. 설치된 위치에서 Cluster Logging Operator 포드를 이동할 수 없습니다.

예를 들어 높은 CPU, 메모리 및 디스크 요구 사항으로 인해 Elasticsearch 포드를 별도의 노드로 이동할 수 있습니다.



#### 참고

최소한 6개의 복제본을 사용하도록 MachineSet를 설정해야 합니다.

#### 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다. 이러한 기능은 기본적으로 설치되지 않습니다.

#### 프로시저

1. **openshift-logging** 프로젝트에서 클러스터 로깅 사용자 정의 리소스를 편집합니다.

```
$ oc edit ClusterLogging instance
```

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
```

```
....
```

```
spec:
  collection:
    logs:
      fluentd:
        resources: null
      type: fluentd
```

```

curation:
  curator:
    nodeSelector: ❶
      node-role.kubernetes.io/infra: "
    resources: null
    schedule: 30 3 * * *
    type: curator
logStore:
  elasticsearch:
    nodeCount: 3
    nodeSelector: ❷
      node-role.kubernetes.io/infra: "
    redundancyPolicy: SingleRedundancy
    resources:
      limits:
        cpu: 500m
        memory: 16Gi
      requests:
        cpu: 500m
        memory: 16Gi
    storage: {}
    type: elasticsearch
  managementState: Managed
  visualization:
    kibana:
      nodeSelector: ❸
        node-role.kubernetes.io/infra: " ❹
      proxy:
        resources: null
      replicas: 1
      resources: null
      type: kibana
....

```

❶ ❷ ❸ ❹ 이동하려는 구성 요소에 적절한 값을 가진 **nodeSelector** 매개변수를 추가합니다. 표시된 형식으로 **nodeSelector**를 사용하거나 노드에 지정된 값에 따라 **<key>:<value>** 쌍을 사용할 수 있습니다.

## 검증 단계

구성 요소가 이동했는지 확인하기 위해 **oc get pod -o wide** 명령을 사용할 수 있습니다.

예를 들면 다음과 같습니다.

- **ip-10-0-147-79.us-east-2.compute.internal** 노드에서 Kibana 포드를 이동하려고 합니다.

```
$ oc get pod kibana-5b8bdf44f9-ccpq9 -o wide
```

## 출력 예

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
NOMINATED NODE READINESS GATES						
kibana-5b8bdf44f9-ccpq9	2/2	Running	0	27s	10.129.2.18	ip-10-0-147-79.us-

```
east-2.compute.internal <none> <none>
```

- Kibana 포드를 전용 인프라 노드인 **ip-10-0-139-48.us-east-2.compute.internal** 노드로 이동하려고 합니다.

```
$ oc get nodes
```

#### 출력 예

NAME	STATUS	ROLES	AGE	VERSION
ip-10-0-133-216.us-east-2.compute.internal	Ready	master	60m	v1.19.0
ip-10-0-139-146.us-east-2.compute.internal	Ready	master	60m	v1.19.0
ip-10-0-139-192.us-east-2.compute.internal	Ready	worker	51m	v1.19.0
ip-10-0-139-241.us-east-2.compute.internal	Ready	worker	51m	v1.19.0
ip-10-0-147-79.us-east-2.compute.internal	Ready	worker	51m	v1.19.0
ip-10-0-152-241.us-east-2.compute.internal	Ready	master	60m	v1.19.0
ip-10-0-139-48.us-east-2.compute.internal	Ready	infra	51m	v1.19.0

노드에는 **node-role.kubernetes.io/infra: "** 레이블이 있습니다.

```
$ oc get node ip-10-0-139-48.us-east-2.compute.internal -o yaml
```

#### 출력 예

```
kind: Node
apiVersion: v1
metadata:
  name: ip-10-0-139-48.us-east-2.compute.internal
  selfLink: /api/v1/nodes/ip-10-0-139-48.us-east-2.compute.internal
  uid: 62038aa9-661f-41d7-ba93-b5f1b6ef8751
  resourceVersion: '39083'
  creationTimestamp: '2020-04-13T19:07:55Z'
  labels:
    node-role.kubernetes.io/infra: "
....
```

- Kibana 포드를 이동하려면 클러스터 로깅 CR을 편집하여 노드 선택기를 추가합니다.

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
....

spec:
....

visualization:
  kibana:
    nodeSelector: ❶
    node-role.kubernetes.io/infra: " ❷
  proxy:
    resources: null
```

```
replicas: 1
resources: null
type: kibana
```

- 1 2 노드 사양의 레이블과 일치하도록 노드 선택기를 추가합니다.

- CR을 저장하면 현재 Kibana 포드가 종료되고 새 포드가 배포됩니다.

```
$ oc get pods
```

#### 출력 예

NAME	READY	STATUS	RESTARTS	AGE
cluster-logging-operator-84d98649c4-zb9g7	1/1	Running	0	29m
elasticsearch-cdm-hwv01pf7-1-56588f554f-kpmlg	2/2	Running	0	28m
elasticsearch-cdm-hwv01pf7-2-84c877d75d-75wqj	2/2	Running	0	28m
elasticsearch-cdm-hwv01pf7-3-f5d95b87b-4nx78	2/2	Running	0	28m
fluentd-42dzz	1/1	Running	0	28m
fluentd-d74rq	1/1	Running	0	28m
fluentd-m5vr9	1/1	Running	0	28m
fluentd-nkxl7	1/1	Running	0	28m
fluentd-pdvqb	1/1	Running	0	28m
fluentd-tflh6	1/1	Running	0	28m
kibana-5b8bdf44f9-ccpq9	2/2	Terminating	0	4m11s
kibana-7d85dcffc8-bfpfp	2/2	Running	0	33s

- 새 포드는 **ip-10-0-139-48.us-east-2.compute.internal** 노드에 있습니다.

```
$ oc get pod kibana-7d85dcffc8-bfpfp -o wide
```

#### 출력 예

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
NOMINATED NODE READINESS GATES						
kibana-7d85dcffc8-bfpfp	2/2	Running	0	43s	10.131.0.22	ip-10-0-139-48.us-east-2.compute.internal
	<none>	<none>				

- 잠시 후 원래 Kibana 포드가 제거됩니다.

```
$ oc get pods
```

#### 출력 예

NAME	READY	STATUS	RESTARTS	AGE
cluster-logging-operator-84d98649c4-zb9g7	1/1	Running	0	30m
elasticsearch-cdm-hwv01pf7-1-56588f554f-kpmlg	2/2	Running	0	29m
elasticsearch-cdm-hwv01pf7-2-84c877d75d-75wqj	2/2	Running	0	29m
elasticsearch-cdm-hwv01pf7-3-f5d95b87b-4nx78	2/2	Running	0	29m
fluentd-42dzz	1/1	Running	0	29m
fluentd-d74rq	1/1	Running	0	29m
fluentd-m5vr9	1/1	Running	0	29m
fluentd-nkxl7	1/1	Running	0	29m



fluentd-pdvqb	1/1	Running	0	29m
fluentd-tflh6	1/1	Running	0	29m
kibana-7d85dcffc8-bfpfp	2/2	Running	0	62s

### 3.9. SYSTEMD-JOURNALD 및 FLUENTD 구성

Fluentd는 저널에서 읽고 저널 기본 설정이 매우 낮기 때문에 저널은 시스템 서비스의 로깅 속도를 유지할 수 없으므로 저널 항목이 손실될 수 있습니다.

저널이 항목을 손실하지 않도록 **RateLimitInterval = 1s** 및 **RateLimitBurst = 10000**(또는 필요한 경우 더 높음)을 설정하는 것이 좋습니다.

#### 3.9.1. 클러스터 로깅을 위한 systemd-journald 구성

프로젝트를 확장할 때 기본 로깅 환경을 조정해야 할 수도 있습니다.

예를 들어, 로그가 누락된 경우 저널에 대한 비율 제한을 늘려야 할 수 있습니다. 클러스터 로깅이 로그를 삭제하지 않고 과도한 리소스를 사용하지 않도록 지정된 기간 동안 보유할 메시지 수를 조정할 수 있습니다.

로그 압축 여부, 로그 보존 기간, 로그 저장 방법 또는 저장 여부 및 기타 설정을 확인할 수도 있습니다.

#### 프로시저

1. 필요한 설정으로 **journald.conf** 파일을 만듭니다.

```
Compress=yes ①
ForwardToConsole=no ②
ForwardToSyslog=no
MaxRetentionSec=1month ③
RateLimitBurst=10000 ④
RateLimitInterval=1s
Storage=persistent ⑤
SyncIntervalSec=1s ⑥
SystemMaxUse=8g ⑦
SystemKeepFree=20% ⑧
SystemMaxFileSize=10M ⑨
```

- ① 로그를 파일 시스템에 쓰기 전에 압축할지 여부를 지정합니다. 메시지를 압축하려면 **yes**를 지정하고 압축하지 않으려면 **no**를 지정합니다. 기본값은 **yes**입니다.
- ② 로그 메시지를 전달할지 여부를 구성합니다. 각각에 대해 기본값은 **no**입니다. 다음을 지정합니다.
  - 시스템 콘솔에 로그를 전달하려면 **ForwardToConsole**을 지정합니다.
  - 로그를 커널 로그 버퍼로 전달하려면 **ForwardToKsmg**를 지정합니다.
  - syslog 데몬으로 전달하려면 **ForwardToSyslog**를 지정합니다.
  - 로그인한 모든 사용자에게 월(wall) 메시지로 메시지를 전달하려면 **ForwardToWall**을 지정합니다.

③

저널 항목을 저장할 최대 시간을 지정합니다. 초를 지정하려면 숫자를 입력합니다. 또는 "year", "month", "week", "day", "h" 또는 "m"과 같은 단위를 포함합니다. 비활성화하려면 **0**을

- 4 속도 제한을 구성합니다. **RateLimitIntervalSec**에서 정의한 시간 간격 동안, **RateLimitBurst**에 지정된 것보다 더 많은 로그를 수신하는 경우 간격이 끝날 때까지 간격 내의 모든 추가 메시지는 삭제됩니다. 기본값인 **RateLimitInterval=1s** 및 **RateLimitBurst=10000**을 설정하는 것이 좋습니다.
- 5 로그 저장 방법을 지정합니다. 기본값은 **persistent**입니다.
  - **/var/log/journal/**에서 메모리에 로그를 저장하기 위한 **volatile**입니다.
  - **/var/log/journal/**의 디스크에 로그를 저장하기 위한 **persistent**입니다. systemd는 디렉토리가 없는 경우 디렉토리를 생성합니다.
  - 디렉토리가 존재하는 경우 **/var/log/journal/**에 로그를 저장하기 위한 **auto**입니다. 존재하지 않는 경우 systemd는 **/run/systemd/journal**에 로그를 임시 저장합니다.
  - 로그를 저장하지 않는 **none**입니다. systemd는 모든 로그를 삭제합니다.
- 6 **ERR, WARNING, NOTICE, INFO** 및 **DEBUG** 로그에 대해 저널 파일을 디스크에 동기화하기 전에 제한 시간을 지정합니다. **CRIT, ALERT** 또는 **EMERG** 로그를 수신하면 systemd가 즉시 동기화됩니다. 기본값은 **1s**입니다.
- 7 저널이 사용할 수 있는 최대 크기를 지정합니다. 기본값은 **8g**입니다.
- 8 시스템에서 사용 가능한 디스크 공간을 지정합니다. 기본값은 **20%**입니다.
- 9 **/var/log/journal**에 지속적으로 저장된 개별 저널 파일의 최대 크기를 지정합니다. 기본값은 **10M**입니다.



#### 참고

속도 제한을 제거하는 경우 이전에 제한되었던 메시지를 처리할 때 시스템 로깅 데몬에서 CPU 사용률이 증가할 수 있습니다.

시스템 설정에 대한 자세한 내용은

<https://www.freedesktop.org/software/systemd/man/journald.conf.html>을 참조하십시오. 해당 페이지에 나열된 기본 설정은 OpenShift Container Platform에 적용되지 않을 수 있습니다.

2. **journald.conf** 파일을 base64로 변환합니다.

```
$ export jrnl_cnf=$( cat /journald.conf | base64 -w0 )
```

3. 마스터 또는 작업자를 위한 새 MachineConfig를 생성하고 **journald.conf** 매개변수를 추가합니다. 예를 들면 다음과 같습니다.

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
name: 50-corp-journald
```

```
spec:
  config:
    ignition:
      version: 3.1.0
    storage:
      files:
      - contents:
          source: data:text/plain;charset=utf-8;base64,{jrnI_cnf}
          mode: 0644 ❶
          overwrite: true
          path: /etc/systemd/journald.conf ❷
```

❶ **journal.conf** 파일에 대한 권한을 설정합니다. **0644** 권한을 설정하는 것이 좋습니다.

❷ base64로 인코딩된 **journal.conf** 파일의 경로를 지정합니다.

4. MachineConfig를 생성합니다.

```
$ oc apply -f <filename>.yaml
```

컨트롤러는 새로운 MachineConfig를 감지하고 새로운 **rendered-worker-<hash>** 버전을 생성합니다.

5. 각 노드에 새로 렌더링된 구성의 롤아웃 상태를 모니터링합니다.

```
$ oc describe machineconfigpool/worker
```

#### 출력 예

```
Name:      worker
Namespace:
Labels:    machineconfiguration.openshift.io/mco-built-in=
Annotations: <none>
API Version: machineconfiguration.openshift.io/v1
Kind:      MachineConfigPool

...

Conditions:
  Message:
  Reason:    All nodes are updating to rendered-worker-
             913514517bcea7c93bd446f4830bc64e
```

## 3.10. 로그 큐레이터 구성

OpenShift Container Platform 4.6에서는 사용자가 정의한 **보존 정책**에 따라 Elasticsearch에서 로그 큐레이션 작업을 수행합니다.

Elasticsearch Curator 도구는 OpenShift Container Platform 4.6 이전의 데이터 모델을 사용하는 Elasticsearch 인덱스를 제거합니다. 이전 데이터에 대한 Curator 인덱스 보존 정책을 수정할 수 있습니다.

### 3.10.1. Curator 일정 구성

클러스터 로깅 설치로 생성된 클러스터 로깅 사용자 정의 리소스를 사용하여 Curator 일정을 지정할 수 있습니다.

#### 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

#### 프로시저

Curator 일정을 구성하려면 다음을 수행합니다.

1. **openshift-logging** 프로젝트에서 클러스터 로깅 사용자 정의 리소스를 편집합니다.

```
$ oc edit clusterlogging instance

apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
...

curation:
  curator:
    schedule: 30 3 * * * 1
    type: curator
```

1. **cron** 형식으로 Curator의 일정을 지정합니다.



#### 참고

시간대는 Curator 포드가 실행되는 호스트 노드를 기준으로 설정됩니다.

### 3.10.2. Curator 인덱스 삭제 구성

OpenShift Container Platform 4.6 이전의 데이터 모델을 사용하는 Elasticsearch 데이터를 삭제하도록 Curator를 구성할 수 있습니다. 프로젝트별 및 전역 설정을 구성할 수 있습니다. 전역 설정은 지정되지 않은 모든 프로젝트에 적용됩니다. 프로젝트별 설정은 전역 설정보다 우선합니다.

#### 사전 요구 사항

- 클러스터 로깅이 설치되어 있어야 합니다.

#### 프로시저

인덱스를 삭제하려면 다음을 수행합니다.

1. OpenShift Container Platform 사용자 정의 Curator 구성 파일을 편집합니다.

```
$ oc edit configmap/curator
```

2. 필요한 대로 다음 매개변수를 설정합니다.

```
config.yaml: |
```

```
project_name:
  action
  unit:value
```

사용 가능한 매개 변수는 다음과 같습니다.

표 3.2. 프로젝트 옵션

변수 이름	설명
<b>project_name</b>	<b>myapp-devel</b> 과 같은 프로젝트의 실제 이름입니다. OpenShift Container Platform <b>작업</b> 로그의 경우 프로젝트 이름으로 이름 <b>.operations</b> 를 사용합니다.
작업	수행할 작업으로, 현재 <b>삭제</b> 만 허용됩니다.
단위	삭제에 사용할 기간으로, <b>일</b> , <b>주</b> 또는 <b>월</b> 이 있습니다.
값	단위 수입니다.

표 3.3. 필터 옵션

변수 이름	설명
<b>.defaults</b>	<b>.defaults</b> 를 <b>project_name</b> 으로 사용하여 지정되지 않은 프로젝트의 기본값을 설정합니다.
<b>.regex</b>	프로젝트 이름과 일치하는 정규식 목록입니다.
패턴	작은 따옴표로 묶은 유효하고 올바르게 이스케이프 된 정규 표현식 패턴입니다.

예를 들어 Curator를 다음과 같이 구성하려면

- **myapp-dev** 프로젝트에서 **1일**이 지난 인덱스 삭제
- **1주**가 지난 **myapp-qe** 프로젝트에서 인덱스 삭제
- **8주**가 지난 **작업** 로그 삭제
- **31일**이 지난 후 다른 모든 프로젝트 인덱스 삭제
- **^project\..+\.dev.\*\$** regex와 일치하는 1일이 지난 인덱스 삭제
- **^project\..+\.test.\*\$** regex와 일치하는 2일이 지난 인덱스 삭제

사용:

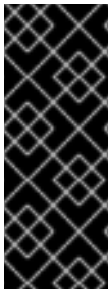
```
config.yaml: |
  .defaults:
    delete:
      days: 31
```

```
.operations:
  delete:
    weeks: 8

myapp-dev:
  delete:
    days: 1

myapp-qe:
  delete:
    weeks: 1

.regex:
  - pattern: '^project\..+\-dev\..*$'
    delete:
      days: 1
  - pattern: '^project\..+\-test\..*$'
    delete:
      days: 2
```



### 중요

작업에 대해 **\$UNIT**으로 **월**을 사용하면 Curator는 현재 달의 현재 날짜가 아니라 현재 달의 첫 날부터 계산을 시작합니다. 예를 들어 오늘이 4월 15일이고 오늘보다 2개월 오래된 인덱스를 삭제하려는 경우(삭제: 월: 2) Curator는 2월 15일 이전인 인덱스를 삭제하지 않습니다. 2월 1일 이전의 인덱스를 삭제합니다. 즉, 현재 달의 첫날로 되돌아간 다음 해당 날짜로부터 전체 두 달 전으로 되돌아갑니다. Curator를 정확하게 사용하려면 일을 사용하는 것이 가장 좋습니다(예: 삭제: 일: 30).

## 3.11. 유지보수 및 지원

### 3.11.1. 지원되지 않는 구성 정보

지원되는 클러스터 로깅 구성 방법은 이 설명서에 설명된 옵션을 사용하여 구성하는 것입니다. 다른 구성은 지원되지 않으므로 사용하지 마십시오. 구성 패러다임은 OpenShift Container Platform 릴리스마다 변경될 수 있으며 이러한 경우는 모든 구성 가능성이 제어되는 경우에만 정상적으로 처리될 수 있습니다. 이 문서에 설명된 것과 다른 구성을 사용하는 경우 Elasticsearch Operator와 Cluster Logging Operator가 차이를 조정하므로 변경 사항이 사라집니다. Operator는 기본적으로 그리고 의도적으로 모든 항목을 정의된 상태로 되돌립니다.



### 참고

OpenShift Container Platform 설명서에 설명되지 않은 구성을 **수행해야 하는** 경우 클러스터 로깅 Operator 또는 Elasticsearch Operator를 **관리되지 않음**으로 **설정해야 합니다**. 관리되지 않는 클러스터 로깅 환경은 **지원되지 않으며** 클러스터 로깅을 **관리 대상**으로 반환할 때까지 업데이트를 받지 않습니다.

### 3.11.2. 지원되지 않는 로깅 구성

다음 구성 요소를 수정하려면 클러스터 로깅 Operator를 관리되지 않는 상태로 설정해야 합니다.

- Curator CronJob

- Elasticsearch CR
- Kibana 배포
- **fluent.conf** 파일
- Fluentd DaemonSet

다음 구성 요소를 수정하려면 Elasticsearch Operator를 관리되지 않는 상태로 설정해야 합니다.

- Elasticsearch 배포 파일.

명시적으로 지원되지 않는 경우는 다음과 같습니다.

- **기본 로그 회전 구성.** 기본 로그 회전 구성을 수정할 수 없습니다.
- **수집된 로그 위치 구성.** 로그 수집기 출력 파일의 위치는 기본적으로 `/var/log/fluentd/fluentd.log`입니다.
- **제한 로그 수집.** 로그 수집기에서 로그를 읽는 속도를 조절할 수 없습니다.
- **로그 수집 JSON 구문 분석 구성.** JSON에서 로그 메시지를 포맷할 수 없습니다.
- **환경 변수를 사용하여 로깅 수집기 구성.** 환경 변수를 사용하여 로그 수집기를 수정할 수 없습니다.
- **로그 수집기에서 로그를 정규화하는 방법 구성.** 기본 로그 정규화를 수정할 수 없습니다.
- **스크립트 배포에서 Curator 구성.** 스크립트 배포에서는 로그 큐레이션을 구성할 수 없습니다.
- **Curator 작업 파일 사용.** Curator ConfigMap을 사용하여 Curator 작업 파일을 수정할 수 없습니다.

### 3.11.3. 관리되지 않는 Operator에 대한 지원 정책

Operator의 *관리 상태*는 Operator가 설계된 대로 클러스터의 관련 구성 요소에 대한 리소스를 적극적으로 관리하고 있는지 판별합니다. Operator가 *관리되지 않는* 상태로 설정되면 구성 변경에 응답하지 않거나 업데이트를 받지 않습니다.

비 프로덕션 클러스터 또는 디버깅 중에 이 기능이 유용할 수 있지만 관리되지 않는 상태의 Operator는 지원되지 않으며 클러스터 관리자는 개별 구성 요소 구성 및 업그레이드를 완전히 제어한다고 가정합니다.

다음 방법을 사용하여 Operator를 관리되지 않는 상태로 설정할 수 있습니다.

- **개별 Operator 구성**

개별 Operator는 구성에 **managementState** 매개변수가 있습니다. Operator에 따라 다른 방식으로 액세스할 수 있습니다. 예를 들어, 클러스터 로깅 Operator는 관리하는 사용자 정의 리소스(CR)를 수정하여 이를 수행하고 클러스터 샘플 Operator는 클러스터 전체 구성 리소스를 사용합니다.

**managementState** 매개변수를 **관리되지 않음**으로 변경하면 Operator가 리소스를 적극적으로 관리하지 않고 관련 구성 요소와 관련된 조치를 수행하지 않습니다. 일부 Operator는 클러스터를 손상시키고 수동 복구가 필요할 수 있으므로 이 관리 상태를 지원하지 않을 수 있습니다.



### 주의

개별 Operator를 **관리되지 않음** 상태로 변경하면 해당 특정 구성 요소 및 기능이 지원되지 않습니다. 지원을 계속하려면 보고된 문제를 **관리** 상태로 재현해야 합니다.

- **Cluster Version Operator(CVO) 재정의**

**spec.overrides** 매개변수를 CVO 구성에 추가하여 관리자가 구성 요소에 대한 CVO 동작에 대한 재정의 목록을 제공할 수 있습니다. 구성 요소에 대해 **spec.overrides[].unmanaged** 매개변수를 **true**로 설정하면 클러스터 업그레이드가 차단되고 CVO 재정의가 설정된 후 관리자에게 경고합니다.

Disabling ownership via cluster version overrides prevents upgrades. Please remove overrides before continuing.



### 주의

CVO 재정의 설정하면 전체 클러스터가 지원되지 않는 상태가 됩니다. 지원을 계속하려면 재정의 제거 후 보고된 문제를 재현해야 합니다.



## 4장. 클러스터 로그 보기

### 4.1. 클러스터 로그 보기

CLI 및 OpenShift Container Platform 웹 콘솔에서 클러스터 로그를 볼 수 있습니다.

#### 4.1.1. 클러스터 로그 보기

CLI에서 클러스터 로그를 볼 수 있습니다.

##### 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

##### 프로시저

클러스터 로그를 보려면 다음을 수행합니다.

**oc logs [-f]<pod\_name>** 명령을 사용합니다. 여기서 **-f**는 선택사항입니다.

```
$ oc logs -f <any-fluentd-pod> -n openshift-logging 1
```

- 1** 로그 수집기 포드의 이름을 지정합니다. **-f** 옵션을 사용하여 로그에 기록되는 내용을 따르십시오.

예를 들면 다음과 같습니다

```
$ oc logs -f fluentd-ht42r -n openshift-logging
```

로그 파일의 내용이 인쇄됩니다.

기본적으로 Fluentd는 로그의 말미 또는 끝에서 로그를 읽습니다.

#### 4.1.2. OpenShift Container Platform 웹 콘솔에서 클러스터 로그 보기

OpenShift Container Platform 웹 콘솔에서 클러스터 로그를 볼 수 있습니다.

##### 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

##### 프로시저

클러스터 로그를 보려면 다음을 수행합니다.

- OpenShift Container Platform 콘솔에서 **워크로드** → **포드**로 이동합니다.
- 드롭다운 메뉴에서 **openshift-logging** 프로젝트를 선택합니다.
- fluentd** 접두사가 있는 로깅 수집기 포드 중 하나를 클릭합니다.
- 로그를 클릭합니다.

기본적으로 Fluentd는 로그의 말미 또는 끝에서 로그를 읽습니다.

## 4.2. KIBANA를 사용하여 클러스터 로그 보기

OpenShift Container Platform 클러스터 로깅에는 수집된 로그 데이터를 시각화하기 위한 웹 콘솔이 포함되어 있습니다. 현재 OpenShift Container Platform은 시각화를 위해 Kibana 콘솔을 배포합니다.

로그 시각화 프로그램을 사용하여 다음을 수행할 수 있습니다.

- **발견** 탭을 사용하여 데이터를 검색하고 찾아봅니다.
- **시각화** 탭을 사용하여 데이터를 차트로 작성하고 매핑합니다.
- **대시보드** 탭을 사용하여 사용자 정의 대시보드를 만들고 봅니다.

Kibana 인터페이스의 사용 및 구성은 이 문서의 범위를 벗어납니다. 인터페이스 사용에 대한 자세한 내용은 [Kibana 문서](#)를 참조하십시오.

### 4.2.1. 로그 시각화 프로그램 시작

OpenShift Container Platform은 Kibana를 로그 시각화 도구로 사용합니다. Kibana는 히스토그램, 선 그래프, 원형 차트, 히트 맵, 기본 제공 지리 공간 지원 및 기타 시각화를 통해 로그를 쿼리, 검색 및 시각화하는 브라우저 기반 콘솔입니다.

#### 사전 요구 사항

- Kibana에서 **인프라** 및 **감사** 인덱스를 나열하려면 사용자에게 **cluster-admin** 역할, **cluster-reader** 역할 또는 두 역할이 모두 있어야 합니다. 기본 **kubeadmin** 사용자에게는 이러한 인덱스를 나열할 수 있는 적절한 권한이 있습니다.  
기본 프로젝트에서 포드와 로그를 볼 수 있다면 이러한 인덱스에 액세스할 수 있어야 합니다. 다음 명령을 사용하여 현재 사용자에게 적절한 권한이 있는지 확인할 수 있습니다.

```
$ oc auth can-i get pods/logs -n default
```

#### 출력 예

```
yes
```




#### 참고

감사 로그는 기본적으로 내부 OpenShift Container Platform Elasticsearch 인스턴스에 저장되지 않습니다. Kibana에서 감사 로그를 보려면 Log Forward API를 사용하여 감사 로그에 대한 **기본** 출력을 사용하는 파이프 라인을 구성해야 합니다.

#### 프로시저

Kibana를 시작하려면 다음을 수행합니다.

1. OpenShift Container Platform 콘솔에서 Application Launcher  를 클릭하고 **로깅**을 선택합니다.
2. OpenShift Container Platform 콘솔에 로그인할 때 사용하는 것과 동일한 자격 증명을 사용하여 로그인합니다.  
Kibana 인터페이스가 시작됩니다.



## 참고

Kibana 콘솔에 **security\_exception** 오류가 발생하여 Kibana 인덱스에 액세스할 수 없는 경우 OAuth 토큰이 만료되었을 수 있습니다. 이 오류가 표시되면 Kibana 콘솔에서 로그아웃한 후 다시 로그인합니다. 그러면 OAuth 토큰이 새로 고쳐지고 인덱스에 액세스할 수 있어야 합니다.

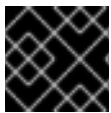
## 5장. 타사 시스템에 로그 전달

기본적으로 OpenShift Container Platform 클러스터 로깅은 클러스터 로깅 사용자 정의 리소스에 정의된 기본 내부 Elasticsearch 로그 저장소로 로그를 보냅니다. 다른 로그 집계에 로그를 전달하려는 경우 OpenShift Container Platform Log Forwarding API를 사용하여 컨테이너, 인프라 및 감사 로그를 클러스터 내부 또는 외부의 특정 엔드 포인트로 보낼 수 있습니다. 다른 유형의 로그를 다른 시스템으로 보내 조직의 각 유형에 액세스할 수 있는 사람을 제어할 수 있습니다. 선택적 TLS 지원을 통해 조직의 요구에 따라 보안 통신을 사용하여 로그를 보낼 수 있습니다.

로그를 외부로 전달할 때 클러스터 로깅 Operator는 Fluentd ConfigMap을 생성하거나 수정하여 원하는 프로토콜을 사용하여 로그를 보냅니다. 외부 로그 집계기에서 프로토콜을 구성해야 합니다.

내부 OpenShift Container Platform Elasticsearch 인스턴스에만 로그를 전달하려면 Log Forwarding API를 구성하지 마십시오.

또는 [Fluentd 전달 프로토콜](#)을 사용하도록 ConfigMap을 생성하거나 외부 시스템에 로그를 전송하기 위해 [syslog 프로토콜](#)을 만들 수 있습니다. 그러나 이러한 로그 전달 방법은 OpenShift Container Platform에서 더 이상 사용되지 않으며 향후 릴리스에서 제거됩니다.



### 중요

동일한 클러스터에서 ConfigMap 방법과 Log Forwarding API를 사용할 수 없습니다.

### 5.1. 타사 시스템으로 로그 전달 정보

클러스터 로그를 외부 타사 시스템으로 전달하려면 OpenShift Container Platform 클러스터 내부 및 외부의 특정 엔드 포인트로 로그를 전송하기 위해 **ClusterLogForwarder** 사용자 정의 리소스(CR)에 지정된 출력 및 파이프라인 조합이 필요합니다. 입력을 사용하여 특정 프로젝트와 관련된 애플리케이션 로그를 엔드 포인트로 전달할 수도 있습니다.

- 출력은 사용자가 정의한 로그 데이터의 대상 또는 로그를 보낼 위치입니다. 출력은 다음 유형 중 하나일 수 있습니다.
  - **elasticsearch.** 외부 Elasticsearch v5.x 또는 v6.x 인스턴스. **elasticsearch** 출력은 TLS 연결을 사용할 수 있습니다.
  - **fluentdForward.** Fluentd를 지원하는 외부 로그 집계 솔루션입니다. 이 옵션은 **Fluentd 전달** 프로토콜을 사용합니다. **fluentForward** 출력은 TCP 또는 TLS 연결을 사용할 수 있으며 비밀에 **shared\_key** 필드를 제공하여 공유 키 인증을 지원합니다. 공유 키 인증은 TLS를 포함하거나 포함하지 않고 사용할 수 있습니다.
  - **syslog.** syslog [RFC3164](#) 또는 [RFC5424](#) 프로토콜을 지원하는 외부 로그 집계 솔루션입니다. **syslog** 출력은 UDP, TCP 또는 TLS 연결을 사용할 수 있습니다.
  - **kafka.** Kafka 브로커. **kafka** 출력은 TCP 또는 TLS 연결을 사용할 수 있습니다.
  - **기본.** 내부 OpenShift Container Platform Elasticsearch 인스턴스입니다. 기본 출력을 구성할 필요는 없습니다. 기본 출력을 구성하는 경우 기본 출력 클러스터 로깅 Operator에 대해 보존되므로 오류 메시지가 나타납니다.

출력 URL 체계에 TLS(HTTPS, TLS 또는 UDPS)가 필요한 경우 TLS 서버측 인증이 활성화됩니다. 또한 클라이언트 인증을 활성화하려면 출력에 **openshift-logging** 프로젝트의 시크릿 이름이 지정되어야 합니다. 시크릿에는 표시되는 각 인증서를 가리키는 **tls.crt**, **tls.key** 및 **ca-bundle.crt** 키가 있어야 합니다.

- **파이프라인**은 한 로그 유형에서 하나 이상의 출력 또는 전송할 로그로의 간단한 라우팅을 정의합니다. 로그 유형은 다음 중 하나입니다.
  - **애플리케이션**. 인프라 컨테이너 애플리케이션을 제외하고 클러스터에서 실행 중인 사용자 애플리케이션에 의해 생성된 컨테이너 로그입니다.
  - **인프라**. **openshift\***, **kube\*** 또는 **기본** 프로젝트에서 실행되는 포드의 컨테이너 로그 및 노드 파일 시스템에서 가져온 저널 로그입니다.
  - **감사**. 노드 감사 시스템(auditd)에서 생성된 로그와 Kubernetes API 서버 및 OpenShift API 서버의 감사 로그입니다.

파이프라인에서 **키:값** 쌍을 사용하여 아웃바운드 로그 메시지에 레이블을 추가할 수 있습니다. 예를 들어 다른 데이터 센터로 전달되는 메시지에 레이블을 추가하거나 유형별로 로그에 레이블을 지정할 수 있습니다. 오브젝트에 추가된 레이블도 로그 메시지와 함께 전달됩니다.

- **입력**은 특정 프로젝트와 관련된 애플리케이션 로그를 파이프라인으로 전달합니다.

파이프 라인에서 **outputRef** 매개변수를 사용하여 로그를 전달하는 위치와 **inputRef** 매개변수를 사용하여 전달하는 로그 유형을 정의합니다.

다음을 확인합니다.

- **ClusterLogForwarder** 오브젝트가 있는 경우 **기본** 출력이 있는 파이프라인이 없으면 로그가 기본 Elasticsearch 인스턴스로 전달되지 않습니다.
- 모든 로그를 내부 OpenShift Container Platform Elasticsearch 인스턴스에만 전달하려면 Log Forwarding API를 구성하지 마십시오.
- 로그 유형에 대한 파이프라인을 정의하지 않으면 정의되지 않은 유형의 로그가 삭제됩니다. 예를 들어 **애플리케이션** 및 **감사** 유형에 대한 파이프라인을 지정하고 **인프라** 유형에 대한 파이프라인을 지정하지 않으면 **인프라** 로그가 삭제됩니다.
- **ClusterLogForwarder** 사용자 정의 리소스(CR)에서 여러 유형의 출력을 사용하여 다른 프로토콜을 지원하는 서버에 로그를 보낼 수 있습니다.
- 내부 OpenShift Container Platform Elasticsearch 인스턴스는 감사 로그를 위한 보안 스토리지를 제공하지 않습니다. 감사 로그를 전달하는 시스템이 조직 및 정부 규정을 준수하고 올바르게 보호 되도록 하는 것이 좋습니다. OpenShift Container Platform 클러스터 로깅은 이러한 규정을 준수하지 않습니다.
- 키 및 비밀, 서비스 계정, 포트 열기 또는 전역 프록시 구성과 같이 외부 대상에 필요할 수 있는 추가 구성을 만들고 유지 관리할 책임이 있습니다.

다음 예제는 감사 로그를 안전한 외부 Elasticsearch 인스턴스로, 인프라 로그를 안전하지 않은 외부 Elasticsearch 인스턴스로, 애플리케이션 로그를 Kafka 브로커로, 애플리케이션 로그를 **my-apps-logs** 프로젝트에서 내부 Elasticsearch 인스턴스로 전달합니다.

### 샘플 로그 전달 출력 및 파이프라인

```
apiVersion: "logging.openshift.io/v1"
kind: ClusterLogForwarder
metadata:
  name: instance ❶
  namespace: openshift-logging ❷
spec:
  outputs:
```

```

- name: elasticsearch-secure ❸
  type: "elasticsearch"
  url: https://elasticsearch.secure.com:9200
  secret:
    name: elasticsearch
- name: elasticsearch-insecure ❹
  type: "elasticsearch"
  url: http://elasticsearch.insecure.com:9200
- name: kafka-app ❺
  type: "kafka"
  url: tls://kafka.secure.com:9093/app-topic
inputs: ❻
- name: my-app-logs
  application:
    namespaces:
      - my-project
pipelines:
- name: audit-logs ❼
  inputRefs:
    - audit
  outputRefs:
    - elasticsearch-secure
    - default
  labels:
    datacenter: east
    secure: true
- name: infrastructure-logs ❽
  inputRefs:
    - infrastructure
  outputRefs:
    - elasticsearch-insecure
  labels:
    datacenter: west
- name: my-app ❾
  inputRefs:
    - my-app-logs
  outputRefs:
    - default
- inputRefs: ❿
  - application
  outputRefs:
    - kafka-app
  labels:
    datacenter: south

```

- ❶ **ClusterLogForwarder** CR의 이름은 **instance**여야 합니다.
- ❷ 로그 전달 CR의 네임스페이스는 **openshift-logging**이어야 합니다.
- ❸ 보안 URL이 있는 보안 시크릿을 사용하여 보안 Elasticsearch 출력을 구성합니다.
  - 출력을 설명하는 이름입니다.
  - 출력 유형: **elasticsearch**.

- 접두사를 포함하여 유효한 절대 URL인 Elasticsearch 인스턴스의 보안 URL 및 포트입니다.
- TLS 통신을 위해 엔드 포인트에 필요한 시크릿입니다. 시크릿은 **openshift-logging** 프로젝트에 있어야 합니다.

**4** 안전하지 않은 Elasticsearch 출력에 대한 구성:

- 출력을 설명하는 이름입니다.
- 출력 유형: **elasticsearch**.
- 접두사를 포함하여 유효한 절대 URL인 Elasticsearch 인스턴스의 안전하지 않은 URL 및 포트입니다.

**5** 보안 URL을 통한 클라이언트 인증 TLS 통신을 사용하는 Kafka 출력 구성

- 출력을 설명하는 이름입니다.
- 출력 유형: **kafka**.
- 접두사를 포함하여 Kafka 브로커의 URL 및 포트를 유효한 절대 URL로 지정합니다.

**6** **my-namespace** 프로젝트에서 애플리케이션 로그를 필터링하기 위한 입력 구성입니다.

**7** 감사 로그를 안전한 외부 Elasticsearch 인스턴스로 전송하기 위한 파이프 라인 구성:

- 선택사항. 파이프라인을 설명하는 이름입니다.
- **inputRefs**는 로그 유형이며 이 예제에서는 **audit**입니다.
- **outputRefs**는 사용할 출력의 이름입니다. 이 예제에서 **elasticsearch-secure**는 보안 Elasticsearch 인스턴스로 전달하고 **기본값**은 내부 Elasticsearch 인스턴스로 전달합니다.
- 선택사항: 로그에 추가할 레이블입니다.

**8** 인프라 로그를 안전하지 않은 외부 Elasticsearch 인스턴스로 전송하는 파이프라인 구성:

**9** **my-project** 프로젝트에서 내부 Elasticsearch 인스턴스로 로그를 전송하기 위한 파이프라인 구성입니다.

- 선택사항. 파이프라인을 설명하는 이름입니다.
- **inputRefs**는 특정 입력인 **my-app-logs**입니다.
- **outputRefs**는 기본값입니다.
- 선택사항: 로그에 추가할 레이블입니다.

**10** 파이프라인 이름 없이 Kafka 브로커에 로그를 전송하는 파이프라인 구성:

- **inputRefs**는 이 예제 애플리케이션에서 로그 유형입니다.
- **outputRefs**는 사용할 출력의 이름입니다.
- 선택사항: 로그에 추가할 레이블입니다.

### 5.1.1. 외부 Elasticsearch 인스턴스로 로그 전달

선택적으로 내부 OpenShift Container Platform Elasticsearch 인스턴스에 추가하거나 대신 외부 Elasticsearch v5.x 또는 v6.x 인스턴스에 로그를 전달할 수 있습니다. OpenShift Container Platform에서 로그를 수신하도록 외부 로그 집계를 구성해야 합니다.

외부 Elasticsearch 인스턴스에 대한 로그 전달을 구성하려면 해당 인스턴스에 대한 출력과 출력을 사용하는 파이프라인이 있는 **ClusterLogForwarder** 사용자 정의 리소스(CR)를 생성합니다. 외부 Elasticsearch 출력은 HTTP(비보안) 또는 HTTPS(보안 HTTP) 연결을 사용할 수 있습니다.

외부 및 내부 Elasticsearch 인스턴스 모두에 로그를 전달하려면 외부 인스턴스에 대한 출력 및 파이프라인과 기본 출력을 사용하여 내부 인스턴스로 로그를 전달하는 파이프라인을 생성합니다. 기본 출력을 생성할 필요가 없습니다. 기본 출력을 구성하는 경우 기본 출력 클러스터 로깅 Operator에 대해 보존되므로 오류 메시지가 나타납니다.



#### 참고

내부 OpenShift Container Platform Elasticsearch 인스턴스에 **만** 로그를 전달하려는 경우 **ClusterLogForwarder** CR을 생성할 필요가 없습니다.

#### 프로시저

1. 다음과 유사한 **ClusterLogForwarder** CR YAML 파일을 생성합니다.

```
apiVersion: "logging.openshift.io/v1"
kind: ClusterLogForwarder
metadata:
  name: instance ①
  namespace: openshift-logging ②
spec:
  outputs:
    - name: elasticsearch-insecure ③
      type: "elasticsearch" ④
      url: http://elasticsearch.insecure.com:9200 ⑤
    - name: elasticsearch-secure
      type: "elasticsearch"
      url: https://elasticsearch.secure.com:9200
      secret:
        name: es-secret ⑥
  pipelines:
    - name: application-logs ⑦
      inputRefs: ⑧
      - application
      - audit
      outputRefs:
        - elasticsearch-secure ⑨
        - default ⑩
      labels:
        logs: application ⑪
    - name: infrastructure-audit-logs ⑫
      inputRefs:
        - infrastructure
      outputRefs:
```



```
- elasticsearch-insecure
labels:
  logs: audit-infra
```

- 1 **ClusterLogForwarder** CR의 이름은 **instance**여야 합니다.
- 2 로그 전달 CR의 네임스페이스는 **openshift-logging**이어야 합니다.
- 3 출력 이름을 지정합니다.
- 4 **elasticsearch** 유형을 지정합니다.
- 5 외부 Elasticsearch 인스턴스의 URL과 포트를 유효한 절대 URL로 지정합니다. **http**(비보안) 또는 **https**(보안 HTTP) 프로토콜을 사용할 수 있습니다. CIDR 주석을 사용하는 클러스터 전체 프록시가 활성화된 경우 출력은 IP 주소가 아닌 서버 이름 또는 FQDN이어야 합니다.
- 6 **https** 접두사를 사용하는 경우 TLS 통신을 위해 엔드 포인트에 필요한 암호 이름을 지정해야 합니다. 시크릿은 **openshift-logging** 프로젝트에 있어야 하며 해당 키가 나타내는 각 인증서를 가리키는 **tls.crt**, **tls.key** 및 **ca-bundle.crt** 키가 있어야 합니다.
- 7 선택사항: 파이프라인의 이름을 지정합니다.
- 8 해당 파이프라인을 사용하여 전달해야 하는 로그 유형을 지정합니다: **애플리케이션**, **인프라** 또는 **감사**.
- 9 로그 전달을 위해 해당 파이프라인과 함께 사용할 출력을 지정합니다.
- 10 선택사항: 기본 출력을 지정하여 내부 Elasticsearch 인스턴스로 로그를 보냅니다.
- 11 선택사항: 로그에 추가할 하나 이상의 레이블입니다.
- 12 선택사항: 지원되는 유형의 다른 외부 로그 집계에 로그를 전달하도록 여러 출력을 구성합니다.
  - 선택사항. 파이프라인을 설명하는 이름입니다.
  - **inputRefs**는 해당 파이프라인을 사용하여 전달할 로그 유형입니다: **application**, **infrastructure** 또는 **audit**.
  - **outputRefs**는 사용할 출력의 이름입니다.
  - 선택사항: 로그에 추가할 하나 이상의 레이블입니다.

2. CR 오브젝트를 생성합니다.

```
$ oc create -f <file-name>.yaml
```

클러스터 로깅 Operator는 Fluentd 포드를 재배포합니다. 포드가 재배포되지 않으면 Fluentd 포드를 삭제하여 강제로 재배포할 수 있습니다.

```
$ oc delete pod --selector logging-infra=fluentd
```

### 5.1.2. Fluentd 정방향 프로토콜을 사용하여 로그 전달

Fluentd 전달 프로토콜을 사용하여 기본 Elasticsearch 로그 저장소 대신 또는 추가로 프로토콜을 수락하도록 구성된 외부 로그 집계로 로그 사본을 보낼 수 있습니다. OpenShift Container Platform에서 로그를 수신하도록 외부 로그 집계를 구성해야 합니다.

전달 프로토콜을 사용하여 로그 전달을 구성하려면 해당 출력을 사용하는 Fluentd 서버 및 파이프라인에 대한 하나 이상의 출력이 있는 **ClusterLogForwarder** 사용자 정의 리소스(CR)를 생성합니다. Fluentd 출력은 TCP(비보안) 또는 TLS(보안 TCP) 연결을 사용할 수 있습니다.



## 참고

또는 ConfigMap을 사용하여 전달 프로토콜을 사용하여 로그를 전달할 수 있습니다. 그러나 이 방법은 OpenShift Container Platform에서 더 이상 사용되지 않으며 향후 릴리스에서 제거됩니다.

## 프로시저

1. 다음과 유사한 **ClusterLogForwarder** CR YAML 파일을 생성합니다.

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: instance ①
  namespace: openshift-logging ②
spec:
  outputs:
    - name: fluentd-server-secure ③
      type: fluentdForward ④
      url: 'tls://fluentdserver.security.example.com:24224' ⑤
      secret: ⑥
        name: fluentd-secret
    - name: fluentd-server-insecure
      type: fluentdForward
      url: 'tcp://fluentdserver.home.example.com:24224'
  pipelines:
    - name: forward-to-fluentd-secure ⑦
      inputRefs: ⑧
        - application
        - audit
      outputRefs:
        - fluentd-server-secure ⑨
        - default ⑩
      labels:
        clusterId: C1234 ⑪
    - name: forward-to-fluentd-insecure ⑫
      inputRefs:
        - infrastructure
      outputRefs:
        - fluentd-server-insecure
      labels:
        clusterId: C1234
```

- ① **ClusterLogForwarder** CR의 이름은 **instance**여야 합니다.
- ② 로그 전달 CR의 네임스페이스는 **openshift-logging**이어야 합니다.

- 3 출력 이름을 지정합니다.
- 4 **fluentdForward** 유형을 지정합니다.
- 5 유효한 절대 URL로 외부 Fluentd 인스턴스의 URL 및 포트를 지정합니다. **tcp**(비보안) 또는 **tls**(보안 TCP) 프로토콜을 사용할 수 있습니다. CIDR 주석을 사용하는 클러스터 전체 프록시가 활성화된 경우 출력은 IP 주소가 아닌 서버 이름 또는 FQDN이어야 합니다.
- 6 **tls** 접두사를 사용하는 경우 TLS 통신을 위해 엔드 포인트에 필요한 시크릿 이름을 지정해야 합니다. 시크릿은 **openshift-logging** 프로젝트에 있어야 하며 해당 키가 나타내는 각 인증서를 가리키는 **tls.crt**, **tls.key** 및 **ca-bundle.crt** 키가 있어야 합니다.
- 7 선택사항. 파이프라인의 이름을 지정합니다.
- 8 해당 파이프라인을 사용하여 전달해야 하는 로그 유형을 지정합니다: **애플리케이션**, **인프라** 또는 **감사**.
- 9 로그 전달을 위해 해당 파이프라인과 함께 사용할 출력을 지정합니다.
- 10 선택사항. 로그를 내부 Elasticsearch 인스턴스로 전달하려면 **기본** 출력을 지정합니다.
- 11 선택사항. 로그에 추가할 하나 이상의 레이블입니다.
- 12 선택사항: 지원되는 유형의 다른 외부 로그 집계에 로그를 전달하도록 여러 출력을 구성합니다.
  - 선택사항. 파이프라인을 설명하는 이름입니다.
  - **inputRefs**는 해당 파이프라인을 사용하여 전달할 로그 유형입니다: **application**, **infrastructure** 또는 **audit**.
  - **outputRefs**는 사용할 출력의 이름입니다.
  - 선택사항: 로그에 추가할 하나 이상의 레이블입니다.

2. CR 오브젝트를 생성합니다.

```
$ oc create -f <file-name>.yaml
```

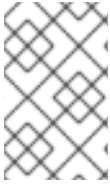
클러스터 로깅 Operator는 Fluentd 포드를 재배포합니다. 포드가 재배포되지 않으면 Fluentd 포드를 삭제하여 강제로 재배포할 수 있습니다.

```
$ oc delete pod --selector logging-infra=fluentd
```

### 5.1.3. syslog 프로토콜을 사용하여 로그 전달

**syslog** [RFC3164](#) 또는 [RFC5424](#) 프로토콜을 사용하여 기본 Elasticsearch 로그 저장소 대신 또는 추가로 프로토콜을 수락하도록 구성된 외부 로그 집계에 로그 사본을 보낼 수 있습니다. OpenShift Container Platform에서 로그를 수신하도록 외부 로그 집계를 구성해야 합니다.

**syslog** 프로토콜을 사용하여 로그 전달을 구성하려면 해당 출력을 사용하는 syslog 서버 및 파이프라인에 대한 하나 이상의 출력이 있는 **ClusterLogForwarder** 사용자 정의 리소스(CR)를 생성합니다. syslog 출력은 UDP, TCP 또는 TLS 연결을 사용할 수 있습니다.



## 참고

또는 ConfigMap을 사용하여 **syslog** RFC3164 프로토콜을 사용하여 로그를 전달할 수 있습니다. 그러나 이 방법은 OpenShift Container Platform에서 더 이상 사용되지 않으며 향후 릴리스에서 제거됩니다.

## 프로시저

1. 다음과 유사한 **ClusterLogForwarder** CR YAML 파일을 생성합니다.

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: instance ❶
  namespace: openshift-logging ❷
spec:
  outputs:
    - name: rsyslog-east ❸
      type: syslog ❹
      syslog: ❺
        facility: local0
        rfc: RFC3164
        payloadKey: message
        severity: informational
      url: 'tls://rsyslogserver.east.example.com:514' ❻
      secret: ❼
        name: syslog-secret
    - name: rsyslog-west
      type: syslog
      syslog:
        appName: myapp
        facility: user
        msgID: mymsg
        procID: myproc
        rfc: RFC5424
        severity: debug
      url: 'udp://rsyslogserver.west.example.com:514'
  pipelines:
    - name: syslog-east ❸
      inputRefs: ❾
        - audit
        - application
      outputRefs: ❿
        - rsyslog-east
        - default ❶❶
      labels:
        syslog: east ❶❷
        secure: true
    - name: syslog-west ❶❸
      inputRefs:
        - infrastructure
      outputRefs:
        - rsyslog-west
```

```
- default
labels:
  syslog: west
```

- 1 **ClusterLogForwarder** CR의 이름은 **instance**여야 합니다.
- 2 로그 전달 CR의 네임스페이스는 **openshift-logging**이어야 합니다.
- 3 출력 이름을 지정합니다.
- 4 **syslog** 유형을 지정합니다.
- 5 선택사항. 아래 나열된 syslog 매개 변수를 지정합니다.
- 6 외부 syslog 인스턴스의 URL 및 포트를 지정합니다. **udp**(비보안), **tcp**(비보안) 또는 **tls**(보안 TCP) 프로토콜을 사용할 수 있습니다. CIDR 주석을 사용하는 클러스터 전체 프록시가 활성화된 경우 출력은 IP 주소가 아닌 서버 이름 또는 FQDN이어야 합니다.
- 7 **tls** 접두사를 사용하는 경우 TLS 통신을 위해 엔드 포인트에 필요한 시크릿 이름을 지정해야 합니다. 시크릿은 **openshift-logging** 프로젝트에 있어야 하며 해당 키가 나타내는 각 인증서를 가리키는 **tls.crt**, **tls.key** 및 **ca-bundle.crt** 키가 있어야 합니다.
- 8 선택사항: 파이프라인의 이름을 지정합니다.
- 9 해당 파이프라인을 사용하여 전달해야 하는 로그 유형을 지정합니다: **애플리케이션**, **인프라** 또는 **감사**.
- 10 로그 전달을 위해 해당 파이프라인과 함께 사용할 출력을 지정합니다.
- 11 선택사항: 기본 출력을 지정하여 내부 Elasticsearch 인스턴스로 로그를 전달합니다.
- 12 선택사항: 로그에 추가할 하나 이상의 레이블입니다.
- 13 선택사항: 지원되는 유형의 다른 외부 로그 집계에 로그를 전달하도록 여러 출력을 구성합니다.
  - 선택사항. 파이프라인을 설명하는 이름입니다.
  - **inputRefs**는 해당 파이프라인을 사용하여 전달할 로그 유형입니다: **application**, **infrastructure** 또는 **audit**.
  - **outputRefs**는 사용할 출력의 이름입니다.
  - 선택사항: 로그에 추가할 하나 이상의 레이블입니다.

2. CR 오브젝트를 생성합니다.

```
$ oc create -f <file-name>.yaml
```

클러스터 로깅 Operator는 Fluentd 포드를 재배포합니다. 포드가 재배포되지 않으면 Fluentd 포드를 삭제하여 강제로 재배포할 수 있습니다.

```
$ oc delete pod --selector logging-infra=fluentd
```

### 5.1.3.1. Syslog 매개변수

**syslog** 출력에 대해 다음을 구성할 수 있습니다. 자세한 내용은 syslog [RFC3164](#) 또는 [RFC5424](#) RFC를 참조하십시오.

- 시설: [syslog 시설](#). 값은 10진수 정수 또는 대소문자를 구분하지 않는 키워드일 수 있습니다.
  - 커널 메시지의 경우 **0** 또는 **kern**
  - 사용자 수준 메시지의 경우 **1** 또는 **user**, 기본값입니다.
  - **2** 또는 **mail** 시스템용 메일
  - 시스템 데몬의 경우 **3** 또는 **daemon**
  - 보안/인증 메시지의 경우 **4** 또는 **auth**
  - syslogd에 의해 내부적으로 생성된 메시지의 경우 **5** 또는 **syslog**
  - 라인 프린터 하위 시스템의 경우 **6** 또는 **lpr**
  - 네트워크 뉴스 서버 시스템의 경우 **7** 또는 **news**
  - UUCP 하위 시스템의 경우 **8** 또는 **uucp**
  - 시계 데몬의 경우 **9** 또는 **cron**
  - 보안 인증 메시지의 경우 **10** 또는 **authpriv**
  - FTP 데몬의 경우 **11** 또는 **ftp**
  - NTP 하위 시스템의 경우 **12** 또는 **ntp**
  - syslog 감사 로그의 경우 **13** 또는 **security**
  - syslog 경고 로그의 경우 **14** 또는 **console**
  - 스케줄링 데몬의 경우 **15** 또는 **solaris-cron**
  - 로컬에서 사용되는 시설의 경우 **16 – 23** 또는 **local0 – local7**
- payloadKey: syslog 메시지에 대한 페이로드로 사용할 레코드 필드입니다.
- rfc: syslog를 사용하여 로그를 보내는 데 사용할 RFC입니다. 기본값은 RFC5424입니다.
- 심각도: 발신되는 syslog 레코드에 설정할 [syslog 심각도](#)입니다. 값은 10진수 정수 또는 대소문자를 구분하지 않는 키워드일 수 있습니다.
  - 시스템을 사용할 수 없음을 나타내는 메시지의 경우 **0** 또는 **Emergency**
  - 조치를 즉시 취해야 함을 나타내는 메시지의 경우 **1** 또는 **Alert**
  - 위험 상태를 나타내는 메시지의 경우 **2** 또는 **Critical**
  - 오류 상태를 나타내는 메시지의 경우 **3** 또는 **Error**
  - 경고 조건을 나타내는 메시지의 경우 **4** 또는 **Warning**
  - 정상이지만 중요한 조건을 나타내는 메시지의 경우 **5** 또는 **Notice**

- 정보성 메시지를 나타내는 메시지의 경우 **6** 또는 **Informational**
- 디버그 수준 메시지를 나타내는 메시지의 경우 **7** 또는 **Debug**, 기본값
- 태그: 태그는 syslog 메시지에서 태그로 사용할 레코드 필드를 지정합니다.
- trimPrefix: 태그에서 지정된 접두사를 제거합니다.

### 5.1.3.2. 추가 RFC5424 syslog 매개변수

RFC5424에는 다음 매개변수가 적용됩니다.

- appName: APP-NAME은 로그를 보낸 애플리케이션을 식별하는 자유 텍스트 문자열입니다. **RFC5424**에 대해 지정해야 합니다.
- msgID: MSGID는 메시지 유형을 식별하는 자유 텍스트 문자열입니다. **RFC5424**에 대해 지정해야 합니다.
- proclD: PROCID는 자유 텍스트 문자열입니다. 값이 변경되면 syslog 보고가 중단되었음을 나타냅니다. **RFC5424**에 대해 지정해야 합니다.

### 5.1.4. Kafka 브로커로 로그 전달

기본 Elasticsearch 로그 저장소에 추가하거나 대신 외부 Kafka 브로커로 로그를 전달할 수 있습니다.

외부 Kafka 인스턴스에 대한 로그 전달을 구성하려면 해당 인스턴스에 대한 출력과 출력을 사용하는 파이프라인이 있는 **ClusterLogForwarder** 사용자 정의 리소스(CR)를 생성합니다. 출력에 특정 Kafka 주제를 포함하거나 기본값을 사용할 수 있습니다. Kafka 출력은 TCP(비보안) 또는 TLS(보안 TCP) 연결을 사용할 수 있습니다.

#### 프로시저

1. 다음과 유사한 **ClusterLogForwarder** CR YAML 파일을 생성합니다.

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: instance ①
  namespace: openshift-logging ②
spec:
  outputs:
    - name: app-logs ③
      type: kafka ④
      url: tls://kafka.example.devlab.com:9093/app-topic ⑤
      secret:
        name: kafka-secret ⑥
    - name: infra-logs
      type: kafka
      url: tls://kafka.devlab2.example.com:9093/infra-topic
    - name: audit-logs
      type: kafka
      url: tls://kafka.qelab.example.com:9093/audit-topic
      secret:
        name: kafka-secret-qe
  pipelines:
```

```

- name: app-topic 7
  inputRefs: 8
  - application
  outputRefs: 9
  - app-logs
  labels:
    logType: application 10
- name: infra-topic 11
  inputRefs:
  - infrastructure
  outputRefs:
  - infra-logs
  labels:
    logType: infra
- name: audit-topic
  inputRefs:
  - audit
  outputRefs:
  - audit-logs
  - default 12
  labels:
    logType: audit

```

- 1 **ClusterLogForwarder** CR의 이름은 **instance**여야 합니다.
- 2 로그 전달 CR의 네임스페이스는 **openshift-logging**이어야 합니다.
- 3 출력 이름을 지정합니다.
- 4 **kafka** 유형을 지정합니다.
- 5 Kafka 브로커의 URL과 포트를 유효한 절대 URL로 지정하고 선택적으로 특정 주제를 사용합니다. **tcp**(비보안) 또는 **tls**(보안 TCP) 프로토콜을 사용할 수 있습니다. CIDR 주석을 사용하는 클러스터 전체 프록시가 활성화된 경우 출력은 IP 주소가 아닌 서버 이름 또는 FQDN이어야 합니다.
- 6 **tls** 접두사를 사용하는 경우 TLS 통신을 위해 엔드 포인트에 필요한 시크릿 이름을 지정해야 합니다. 시크릿은 **openshift-logging** 프로젝트에 있어야 하며 해당 키가 나타내는 각 인증서를 가리키는 **tls.crt**, **tls.key** 및 **ca-bundle.crt** 키가 있어야 합니다.
- 7 선택사항: 파이프라인의 이름을 지정합니다.
- 8 해당 파이프라인을 사용하여 전달해야 하는 로그 유형을 지정합니다: **애플리케이션**, **인프라** 또는 **감사**.
- 9 로그 전달을 위해 해당 파이프라인과 함께 사용할 출력을 지정합니다.
- 10 선택사항: 로그에 추가할 하나 이상의 레이블입니다.
- 11 선택사항: 지원되는 유형의 다른 외부 로그 집계에 로그를 전달하도록 여러 출력을 구성합니다.
  - 선택사항. 파이프라인을 설명하는 이름입니다.
  - **inputRefs**는 해당 파이프라인을 사용하여 전달할 로그 유형입니다: **application**, **infrastructure** 또는 **audit**.



- **outputRefs**는 사용할 출력의 이름입니다.
- 선택사항: 로그에 추가할 하나 이상의 레이블입니다.

**12** 선택사항: 내부 Elasticsearch 인스턴스로 로그를 전달하려면 **기본값**을 지정합니다.

2. CR 오브젝트를 생성합니다.

```
$ oc create -f <file-name>.yaml
```

클러스터 로깅 Operator는 Fluentd 포드를 재배포합니다. 포드가 재배포되지 않으면 Fluentd 포드를 삭제하여 강제로 재배포할 수 있습니다.

```
$ oc delete pod --selector logging-infra=fluentd
```

### 5.1.5. 레거시 **Fluentd** 방법을 사용하여 로그 전달

Fluentd **전달** 프로토콜을 사용하여 구성 파일 및 ConfigMap을 생성하여 기본 Elasticsearch 로그 저장소 대신 OpenShift Container Platform 클러스터 외부의 대상으로 로그를 보낼 수 있습니다. OpenShift Container Platform에서 로그를 수신하도록 외부 로그 집계를 구성해야 합니다.



#### 중요

이 로그 전달 방법은 OpenShift Container Platform에서 더 이상 사용되지 않으며 향후 릴리스에서 제거됩니다.

Fluentd **전달** 프로토콜을 사용하여 로그를 보내려면 외부 로그 수집기를 가리키는 **secure-forward.conf**라는 구성 파일을 만듭니다. 그런 다음 해당 파일을 사용하여 **openshift-logging** 네임스페이스에서 **secure-forward**라는 ConfigMap을 생성합니다. OpenShift Container Platform은 로그를 전달할 때 사용합니다.

#### 샘플 **Fluentd** 구성 파일

```
<store>
  @type forward
  <security>
    self_hostname fluentd.example.com
    shared_key "fluent-receiver"
  </security>
  transport tls
  tls_verify_hostname false
  tls_cert_path '/etc/ocp-forward/ca-bundle.crt'
  <buffer>
    @type file
    path '/var/lib/fluentd/secureforwardlegacy'
    queued_chunks_limit_size "1024"
    chunk_limit_size "1m"
    flush_interval "5s"
    flush_at_shutdown "false"
    flush_thread_count "2"
    retry_max_interval "300"
    retry_forever true
    overflow_action "exception"
```

```

</buffer>
<server>
  host fluent-receiver.example.com
  port 24224
</server>
</store>

```

## 프로시저

레거시 Fluentd 방법을 사용하여 로그를 전달하도록 OpenShift Container Platform을 구성하려면 다음을 수행하십시오.

1. **secure-forward**라는 구성 파일을 생성하고 **<store>** 스탠자에서 다음과 유사한 매개변수를 지정합니다.

```

<store>
  @type forward
  <security>
    self_hostname <common-name> ❶
    shared_key <key> ❷
  </security>
  transport tls ❸
  tls_verify_hostname <value> ❹
  tls_cert_path <path_to_file> ❺
  <buffer> ❻
    @type file
    path '/var/lib/fluentd/secureforwardlegacy'
    queued_chunks_limit_size "#{ENV['BUFFER_QUEUE_LIMIT'] || '1024' }"
    chunk_limit_size "#{ENV['BUFFER_SIZE_LIMIT'] || '1m' }"
    flush_interval "#{ENV['FORWARD_FLUSH_INTERVAL'] || '5s' }"
    flush_at_shutdown "#{ENV['FLUSH_AT_SHUTDOWN'] || 'false' }"
    flush_thread_count "#{ENV['FLUSH_THREAD_COUNT'] || 2}"
    retry_max_interval "#{ENV['FORWARD_RETRY_WAIT'] || '300' }"
    retry_forever true
    overflow_action "#{ENV['BUFFER_QUEUE_FULL_ACTION'] || 'exception' }"
  </buffer>
  <server>
    name ❼
    host ❽
    hostlabel ❾
    port ❿
  </server>
  <server> 11
    name
    host
  </server>

```

- ❶ 자동 생성된 인증서 공통 이름(CN)의 기본값을 지정합니다.
- ❷ 노드 간 공유 키 입력
- ❸ TLS 유효성 검증을 사용하려면 **tls**를 지정합니다.
- ❹ 서버 인증서 호스트 이름을 확인하려면 **true**로 설정합니다. 서버 인증서 호스트 이름을 무시하려면 **false**로 설정합니다.

5. 사설 CA 인증서 파일의 경로를 `/etc/ocp-forward/ca_cert.pem`으로 지정합니다.
6. 필요에 따라 `Fluentd 버퍼 매개변수`를 지정합니다.
7. 선택적으로 이 서버의 이름을 입력합니다.
8. 서버의 호스트 이름 또는 IP를 지정합니다.
9. 서버의 호스트 레이블을 지정합니다.
10. 서버의 포트를 지정합니다.
11. 선택적으로 추가 서버를 추가합니다. 둘 이상의 서버를 지정하면 **forward**는 이러한 서버 노드를 라운드 로빈 순서로 사용합니다.

상호 TLS(mTLS) 인증을 사용하려면 [Fluentd 설명서](#)에서 클라이언트 인증서, 키 매개변수 및 기타 설정에 대한 정보를 참조하십시오.

2. 구성 파일에서 **openshift-logging** 네임스페이스에 **secure-forward**라는 ConfigMap을 생성합니다.

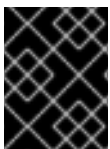
```
$ oc create configmap secure-forward --from-file=secure-forward.conf -n openshift-logging
```

클러스터 로깅 Operator는 Fluentd 포드를 재배포합니다. 포드가 재배포되지 않으면 Fluentd 포드를 삭제하여 강제로 재배포할 수 있습니다.

```
$ oc delete pod --selector logging-infra=fluentd
```

### 5.1.6. 레거시 **syslog** 방법을 사용하여 로그 전달

**syslog** RFC3164 프로토콜을 사용하여 구성 파일 및 ConfigMap을 생성하여 기본 Elasticsearch 로그 저장소 대신 OpenShift Container Platform 클러스터 외부의 대상으로 로그를 보낼 수 있습니다. OpenShift Container Platform에서 로그를 수신하도록 외부 로그 집계를 구성해야 합니다.



#### 중요

이 로그 전달 방법은 OpenShift Container Platform에서 더 이상 사용되지 않으며 향후 릴리스에서 제거됩니다.

**syslog** 프로토콜에는 두 가지 버전이 있습니다.

- **out\_syslog**: UDP를 통해 통신하는 버퍼되지 않은 구현은 데이터를 버퍼링하지 않고 결과를 즉시 기록합니다.
- **out\_syslog\_buffered**: TCP를 통해 통신하고 데이터를 **체크로 버퍼링** 하는 버퍼링된 구현입니다.

**syslog** 프로토콜을 사용하여 로그를 보내려면 로그를 전달하는 데 필요한 정보와 함께 **syslog.conf**라는 구성 파일을 만듭니다. 그런 다음 해당 파일을 사용하여 OpenShift Container Platform이 로그를 전달할 때 사용하는 **openshift-logging** 네임스페이스에 **syslog**라는 ConfigMap을 만듭니다.

#### 샘플 **syslog** 구성 파일

```
<store>
@type syslog_buffered
```

```
remote_syslog rsyslogserver.example.com
port 514
hostname fluentd-4nzfz
remove_tag_prefix tag
facility local0
severity info
use_record true
payload_key message
rfc 3164
</store>
```

다음 **syslog** 매개변수를 구성할 수 있습니다. 자세한 내용은 [RFC3164](#) 에서 참조하십시오.

- 시설: **syslog facility**. 값은 10진수 정수 또는 대소문자를 구분하지 않는 키워드일 수 있습니다.
  - 커널 메시지의 경우 **0** 또는 **kern**
  - 사용자 수준 메시지의 경우 **1** 또는 **user**, 기본값입니다.
  - 메일 시스템의 경우 **2** 또는 **mail**
  - 시스템 데몬의 경우 **3** 또는 **daemon**
  - 보안/인증 메시지의 경우 **4** 또는 **auth**
  - syslogd에 의해 내부적으로 생성된 메시지의 경우 **5** 또는 **syslog**
  - 라인 프린터 하위 시스템의 경우 **6** 또는 **lpr**
  - 네트워크 뉴스 서브 시스템의 경우 **7** 또는 **news**
  - UUCP 하위 시스템의 경우 **8** 또는 **uucp**
  - 시계 데몬의 경우 **9** 또는 **cron**
  - 보안 인증 메시지의 경우 **10** 또는 **authpriv**
  - FTP 데몬의 경우 **11** 또는 **ftp**
  - NTP 하위 시스템의 경우 **12** 또는 **ntp**
  - syslog 감사 로그의 경우 **13** 또는 **security**
  - syslog 경고 로그의 경우 **14** 또는 **console**
  - 스케줄링 데몬의 경우 **15** 또는 **solaris-cron**
  - 로컬에서 사용되는 시설의 경우 **16 - 23** 또는 **local0 - local7**
- payloadKey: syslog 메시지에 대한 페이로드로 사용할 레코드 필드입니다.
- rfc: syslog를 사용하여 로그를 보내는 데 사용할 RFC입니다.
- 심각도: 발신되는 syslog 레코드에 설정할 **syslog 심각도**입니다. 값은 10진수 정수 또는 대소문자를 구분하지 않는 키워드일 수 있습니다.
  - 시스템을 사용할 수 없음을 나타내는 메시지의 경우 **0** 또는 **Emergency**

- 조치를 즉시 취해야 함을 나타내는 메시지의 경우 **1** 또는 **Alert**
  - 위험 상태를 나타내는 메시지의 경우 **2** 또는 **Critical**
  - 오류 상태를 나타내는 메시지의 경우 **3** 또는 **Error**
  - 경고 조건을 나타내는 메시지의 경우 **4** 또는 **Warning**
  - 정상이지만 중요한 조건을 나타내는 메시지의 경우 **5** 또는 **Notice**
  - 정보성 메시지를 나타내는 메시지의 경우 **6** 또는 **Informational**
  - 디버그 수준 메시지를 나타내는 메시지의 경우 **7** 또는 **Debug**, 기본값
- 태그: syslog 메시지에서 태그로 사용할 레코드 필드입니다.
  - trimPrefix: 태그에서 제거할 접두사입니다.

## 프로시저

레거시 구성 방법을 사용하여 로그를 전달하도록 OpenShift Container Platform을 구성하려면 다음을 수행하십시오.

1. **syslog.conf**라는 구성 파일을 만들고 **<store>** 스탠자에서 다음과 유사한 매개변수를 지정합니다.

```
<store>
@type <type> 1
remote_syslog <syslog-server> 2
port 514 3
hostname <host> 4
remove_tag_prefix <prefix> 5
facility <value>
severity <value>
use_record <value>
payload_key message
rfc 3164 6
</store>
```

1. 사용할 프로토콜을 지정하십시오(**syslog** 또는 **syslog\_buffered**).
2. syslog 서버의 FQDN 또는 IP 주소를 지정합니다.
3. syslog 서버의 포트를 지정합니다.
4. 이 syslog 서버의 이름을 지정합니다.
5. 선택사항. 적절한 syslog 매개변수를 지정합니다. 예를 들면 다음과 같습니다.
  - syslog 접두사에서 지정된 **태그** 필드를 제거하기 위한 매개변수입니다.
  - 지정된 필드를 syslog 키로 설정하는 매개변수입니다.
  - syslog 로그 기능 또는 소스를 지정하는 매개변수입니다.
  - syslog 로그 심각도를 지정하는 매개변수입니다.

- 가능한 경우 레코드에서 심각도 및 기능을 사용하는 매개변수입니다. **true**인 경우 **container\_name, namespace\_name, pod\_name**이 출력 콘텐츠에 포함됩니다.
- syslog 메시지의 페이로드를 설정하기 위한 키를 지정하는 매개변수입니다. 기본값은 **message**입니다.

**6** 사용할 RFC 프로토콜을 지정합니다: **3164**.

2. 구성 파일에서 **openshift-logging** 네임스페이스에 **syslog**라는 ConfigMap을 생성합니다.

```
$ oc create configmap syslog --from-file=syslog.conf -n openshift-logging
```

클러스터 로깅 Operator는 Fluentd 포드를 재배포합니다. 포드가 재배포되지 않으면 Fluentd 포드를 삭제하여 강제로 재배포할 수 있습니다.

```
$ oc delete pod --selector logging-infra=fluentd
```

## 6장. KUBERNETES 이벤트 수집 및 저장

OpenShift Container Platform 이벤트 라우터는 Kubernetes 이벤트를 감시하고 클러스터 로깅별 수집을 위해 로깅하는 포드입니다. 이벤트 라우터를 수동으로 배포해야 합니다.

이벤트 라우터는 모든 프로젝트에서 이벤트를 수집하여 **STDOUT**에 작성합니다. Fluentd는 이러한 이벤트를 수집하여 OpenShift Container Platform Elasticsearch 인스턴스로 전달합니다. Elasticsearch는 이벤트를 **인프라** 인덱스에 인덱싱합니다.



### 중요

이벤트 라우터는 Fluentd에 추가 로드를 추가하고 처리할 수 있는 다른 로그 메시지 수에 영향을 미칠 수 있습니다.

### 6.1. 이벤트 라우터 배포 및 구성

다음 단계를 사용하여 이벤트 라우터를 클러스터에 배포합니다. 항상 이벤트 라우터를 **openshift-logging** 프로젝트에 배포하여 클러스터 전체에서 이벤트를 수집해야 합니다.

다음 템플릿 오브젝트는 이벤트 라우터에 필요한 서비스 계정, 클러스터 역할 및 클러스터 역할 바인딩을 만듭니다. 템플릿은 또한 이벤트 라우터 포드를 구성하고 배포합니다. 변경하지 않고 이 템플릿을 사용하거나 배포 오브젝트 CPU 및 메모리 요청을 변경할 수 있습니다.

#### 사전 요구 사항

- 서비스 계정을 만들고 클러스터 역할 바인딩을 업데이트하려면 적절한 권한이 필요합니다. 예를 들어 **cluster-admin** 역할이 있는 사용자로 다음 템플릿을 실행할 수 있습니다.
- 클러스터 로깅이 설치되어 있어야 합니다.

#### 프로시저

1. 이벤트 라우터에 대한 템플릿을 만듭니다.

```
kind: Template
apiVersion: v1
metadata:
  name: eventrouter-template
  annotations:
    description: "A pod forwarding kubernetes events to cluster logging stack."
    tags: "events,EFK,logging,cluster-logging"
objects:
  - kind: ServiceAccount 1
    apiVersion: v1
    metadata:
      name: eventrouter
      namespace: ${NAMESPACE}
  - kind: ClusterRole 2
    apiVersion: v1
    metadata:
      name: event-reader
    rules:
      - apiGroups: [""]
        resources: ["events"]
```

```

  verbs: ["get", "watch", "list"]
- kind: ClusterRoleBinding 3
  apiVersion: v1
  metadata:
    name: event-reader-binding
  subjects:
- kind: ServiceAccount
  name: eventrouter
  namespace: ${NAMESPACE}
  roleRef:
    kind: ClusterRole
    name: event-reader
- kind: ConfigMap 4
  apiVersion: v1
  metadata:
    name: eventrouter
    namespace: ${NAMESPACE}
  data:
    config.json: |-
      {
        "sink": "stdout"
      }
- kind: Deployment 5
  apiVersion: apps/v1
  metadata:
    name: eventrouter
    namespace: ${NAMESPACE}
  labels:
    component: eventrouter
    logging-infra: eventrouter
    provider: openshift
  spec:
    selector:
      matchLabels:
        component: eventrouter
        logging-infra: eventrouter
        provider: openshift
    replicas: 1
    template:
      metadata:
        labels:
          component: eventrouter
          logging-infra: eventrouter
          provider: openshift
        name: eventrouter
      spec:
        serviceAccount: eventrouter
        containers:
- name: kube-eventrouter
  image: ${IMAGE}
  imagePullPolicy: IfNotPresent
  resources:
    requests:
      cpu: ${CPU}
      memory: ${MEMORY}
  volumeMounts:

```



```

      - name: config-volume
        mountPath: /etc/eventrouter
    volumes:
      - name: config-volume
        configMap:
          name: eventrouter
  parameters:
    - name: IMAGE
      displayName: Image
      value: "registry.redhat.io/openshift4/ose-logging-eventrouter:latest"
    - name: CPU ⑥
      displayName: CPU
      value: "100m"
    - name: MEMORY ⑦
      displayName: Memory
      value: "128Mi"
    - name: NAMESPACE
      displayName: Namespace
      value: "openshift-logging" ⑧

```

- ① 이벤트 라우터에 대한 **openshift-logging** 프로젝트에서 서비스 계정을 만듭니다.
- ② 클러스터의 이벤트를 모니터링할 ClusterRole을 만듭니다.
- ③ ClusterRole을 ServiceAccount에 바인딩하는 ClusterRoleBinding을 만듭니다.
- ④ **openshift-logging** 프로젝트에서 ConfigMap을 생성하여 필요한 **config.json** 파일을 생성합니다.
- ⑤ **openshift-logging** 프로젝트에서 배포를 생성하여 이벤트 라우터 포드를 생성하고 구성합니다.
- ⑥ 이벤트 라우터 포드에 할당할 최소 메모리 양을 지정합니다. 기본값은 **128Mi**입니다.
- ⑦ 이벤트 라우터 포드에 할당할 최소 CPU 양을 지정합니다. 기본값은 **100m**입니다.
- ⑧ 오브젝트를 설치할 **openshift-logging** 프로젝트를 지정합니다.

2. 다음 명령을 사용하여 템플릿을 처리하고 적용합니다.

```
$ oc process -f <templatefile> | oc apply -f -
```

예를 들면 다음과 같습니다.

```
$ oc process -f eventrouter.yaml | oc apply -f -
```

#### 출력 예

```

serviceaccount/logging-eventrouter created
clusterrole.authorization.openshift.io/event-reader created
clusterrolebinding.authorization.openshift.io/event-reader-binding created
configmap/logging-eventrouter created
deployment.apps/logging-eventrouter created

```

### 3. openshift-logging 프로젝트에 이벤트 라우터가 설치되었는지 확인합니다.

#### a. 새 이벤트 라우터 포드 보기:

```
$ oc get pods --selector component=eventrouter -o name -n openshift-logging
```

#### 출력 예

```
pod/cluster-logging-eventrouter-d649f97c8-qvv8r
```

#### b. 이벤트 라우터에서 수집한 이벤트 보기:

```
$ oc logs <cluster_logging_eventrouter_pod> -n openshift-logging
```

예를 들면 다음과 같습니다.

```
$ oc logs cluster-logging-eventrouter-d649f97c8-qvv8r -n openshift-logging
```

#### 출력 예

```
{
  "verb": "ADDED",
  "event": {
    "metadata": {
      "name": "openshift-service-catalog-controller-manager-remover.1632d931e88fcd8f",
      "namespace": "openshift-service-catalog-removed",
      "selfLink": "/api/v1/namespaces/openshift-service-catalog-removed/events/openshift-service-catalog-controller-manager-remover.1632d931e88fcd8f",
      "uid": "787d7b26-3d2f-4017-b0b0-420db4ae62c0",
      "resourceVersion": "21399",
      "creationTimestamp": "2020-09-08T15:40:26Z"
    },
    "involvedObject": {
      "kind": "Job",
      "namespace": "openshift-service-catalog-removed",
      "name": "openshift-service-catalog-controller-manager-remover",
      "uid": "fac9f479-4ad5-4a57-8adc-cb25d3d9cf8f",
      "apiVersion": "batch/v1",
      "resourceVersion": "21280"
    },
    "reason": "Completed",
    "message": "Job completed",
    "source": {
      "component": "job-controller"
    },
    "firstTimestamp": "2020-09-08T15:40:26Z",
    "lastTimestamp": "2020-09-08T15:40:26Z",
    "count": 1,
    "type": "Normal"
  }
}
```

Elasticsearch **인프라** 인덱스를 사용하는 인덱스 패턴을 생성하여 이벤트를 보도록 Kibana을 사용할 수도 있습니다.

## 7장. 클러스터 로깅 배포

OpenShift Container Platform 클러스터를 4.4에서 4.5로 업데이트한 후 Elasticsearch Operator 및 클러스터 로깅 Operator를 4.4에서 4.5로 업데이트할 수 있습니다.

클러스터 로깅 4.5에는 새로운 Elasticsearch 버전인 Elasticsearch 6.8.1과 강화된 보안 플러그인인 OpenDistro for Elasticsearch가 도입되었습니다. 새로운 Elasticsearch 버전은 Elasticsearch 데이터가 인프라, 애플리케이션 및 감사 유형별로만 인덱싱되는 새로운 Elasticsearch 데이터 모델을 도입합니다. 이전에는 데이터가 유형(인프라 및 애플리케이션) 및 프로젝트별로 인덱싱되었습니다.



### 중요

새로운 데이터 모델로 인해 업데이트는 기존 사용자 정의 Kibana 인덱스 패턴 및 시각화를 새 버전으로 마이그레이션하지 않습니다. 업데이트 후 새 인덱스와 일치하도록 Kibana 인덱스 패턴 및 시각화를 다시 생성해야 합니다.

이러한 변경의 특성으로 인해 클러스터 로깅을 4.5로 업데이트할 필요가 없습니다. 그러나 OpenShift Container Platform 4.6으로 업데이트할 때 클러스터 로깅을 4.6으로 업데이트해야 합니다.

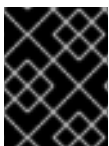
### 7.1. 클러스터 로깅 배포

OpenShift Container Platform 클러스터를 업데이트한 후 Elasticsearch Operator 및 Cluster Logging Operator에 대한 구독을 변경하여 클러스터 로깅을 4.5에서 4.6으로 업데이트할 수 있습니다.

업데이트하는 경우:

- 클러스터 로깅 Operator를 업데이트하기 전에 Elasticsearch Operator를 업데이트해야 합니다.
- Elasticsearch Operator와 Cluster Logging Operator를 모두 업데이트해야 합니다. Elasticsearch Operator가 업데이트되었지만 클러스터 로깅 Operator가 업데이트되지 않은 경우 Kibana를 사용할 수 없습니다.

Elasticsearch Operator 전에 클러스터 로깅 Operator를 업데이트하면 Kibana가 업데이트되지 않고 Kibana 사용자 정의 리소스(CR)가 생성되지 않습니다. 이 문제를 해결하려면 클러스터 로깅 Operator 포드를 삭제합니다. 클러스터 로깅 Operator 포드가 재배포되면 Kibana CR이 생성됩니다.



### 중요

클러스터 로깅 버전이 4.5 이전인 경우 4.6으로 업데이트하기 전에 클러스터 로깅을 4.5로 업그레이드해야 합니다.

#### 사전 요구 사항

- OpenShift Container Platform 클러스터를 4.5에서 4.6으로 업데이트합니다.
- 클러스터 로깅 상태가 정상인지 확인하십시오.
  - 모든 포드가 준비되었습니다.
  - Elasticsearch 클러스터는 정상입니다.
- Elasticsearch 및 Kibana 데이터를 백업합니다.

## 프로시저

1. Elasticsearch Operator를 업데이트합니다.
  - a. 웹 콘솔에서 **Operators → 설치된 Operator**를 클릭합니다.
  - b. **openshift-operators-redhat** 프로젝트를 선택합니다.
  - c. **Elasticsearch Operator**를 클릭합니다.
  - d. 구독 → 채널을 클릭합니다.
  - e. 구독 업데이트 채널 변경 창에서 **4.6**을 선택하고 **저장**을 클릭합니다.
  - f. 몇 초 정도 기다린 후 **Operators → 설치된 Operator**를 클릭합니다.  
Elasticsearch Operator는 4.6으로 표시됩니다. 예를 들면 다음과 같습니다.

```
Elasticsearch Operator
4.6.0-202007012112.p0 provided
by Red Hat, Inc
```

상태 필드가 성공으로 표시될 때까지 기다립니다.

2. 클러스터 로깅 Operator 업데이트:
  - a. 웹 콘솔에서 **Operators → 설치된 Operator**를 클릭합니다.
  - b. **openshift-logging** 프로젝트를 선택합니다.
  - c. 클러스터 로깅 **Operator**를 클릭합니다.
  - d. 구독 → 채널을 클릭합니다.
  - e. 구독 업데이트 채널 변경 창에서 **4.6**을 선택하고 **저장**을 클릭합니다.
  - f. 몇 초 정도 기다린 후 **Operators → 설치된 Operator**를 클릭합니다.  
클러스터 로깅 Operator는 4.6으로 표시됩니다. 예를 들면 다음과 같습니다.

```
Cluster Logging
4.6.0-202007012112.p0 provided
by Red Hat, Inc
```

상태 필드가 성공으로 표시될 때까지 기다립니다.

3. 로깅 구성 요소를 확인하십시오.
  - a. 모든 Elasticsearch 포드가 **준비** 상태인지 확인하십시오.

```
$ oc get pod -n openshift-logging --selector component=elasticsearch
```

## 출력 예

NAME	READY	STATUS	RESTARTS	AGE
elasticsearch-cdm-1pbrl44l-1-55b7546f4c-mshhk	2/2	Running	0	31m
elasticsearch-cdm-1pbrl44l-2-5c6d87589f-gx5hk	2/2	Running	0	30m
elasticsearch-cdm-1pbrl44l-3-88df5d47-m45jc	2/2	Running	0	29m

- b. Elasticsearch 클러스터가 정상인지 확인하십시오.

```
$ oc exec -n openshift-logging -c elasticsearch elasticsearch-cdm-1pbrl44l-1-55b7546f4c-mshhk -- es_cluster_health
```

```
{
  "cluster_name" : "elasticsearch",
  "status" : "green",
}
....
```

- c. Elasticsearch CronJobs가 생성되었는지 확인하십시오.

```
$ oc project openshift-logging
```

```
$ oc get CronJob elasticsearch-*
```

NAME	SCHEDULE	SUSPEND	ACTIVE	LAST SCHEDULE	AGE
elasticsearch-delete-app	*/15 * * * *	False	0	<none>	27s
elasticsearch-delete-audit	*/15 * * * *	False	0	<none>	27s
elasticsearch-delete-infra	*/15 * * * *	False	0	<none>	27s
elasticsearch-rollover-app	*/15 * * * *	False	0	<none>	27s
elasticsearch-rollover-audit	*/15 * * * *	False	0	<none>	27s
elasticsearch-rollover-infra	*/15 * * * *	False	0	<none>	27s

- d. 로그 저장소가 4.6으로 업데이트되고 인덱스가 녹색인지 확인합니다.

```
$ oc exec -c elasticsearch <any_es_pod_in_the_cluster> -- indices
```

**app-0000x, infra-0000x, audit-0000x, .security** 인덱스가 표시되어야 합니다.

```
Tue Jun 30 14:30:54 UTC 2020
health status index                                uuid                                pri rep
docs.count docs.deleted store.size pri.store.size
green open  infra-000008
bnBvUFEXTWi92z3zWAZieQ 3 1    222195      0    289        144
green open  infra-000004
rtDSzoqsSl6saisSK7Au1Q
3 1    226717      0    297        148
green open  infra-000012
RSf_kUwDSR2xEuKRZMPqZQ 3 1    227623      0    295        147
green open  .kibana_7
1SJdCqIzTPWIIAaOUd78yg 1 1      4          0    0          0
green open  .operations.2020.06.30
aOHMYOa3S_69NJFh2t3yrQ 3 1    4206118     0    8998       4499
green open  project.local-storage.d5c8a3d6-30a3-4512-96df-67c537540072.2020.06.30
O_Uldg2wS5K_L6FyqWxOZg 3 1    91052       0    135        67
green open  infra-000010
iXwL3bnqTuGEABbUDa6OVw 3 1    248368     0    317        158
green open  .searchguard
rQhAbWuLQ9iuTsZeHi_2ew 1 1      5          64    0          0
green open  infra-000009
YN9EsULWSNaxWeeNvOs0RA 3 1    258799     0    337        168
green open  infra-000014
```

```

YP0U6R7FQ_GVQVQZ6Yh9lg 3 1 223788 0 292 146
green open infra-000015
JRBbAbEmSMqK5X40df9HbQ 3 1 224371 0 291 145
green open .orphaned.2020.06.30
n_xQC2dWQzConkvQqei3YA 3 1 9 0 0 0
green open infra-000007
llkkAVSzSOMosWTSAJM_hg 3 1 228584 0 296 148
green open infra-000005
d9BoGQdiQASsS3BBFm2iRA 3 1 227987 0 297 148
green open .kibana.647a750f1787408bf50088234ec0edd5a6a9b2ac
l911Z8dSI23py6GDtyJrA 1 1 5 4 0 0
green open project.ui.29cb9680-864d-43b2-a6cf-134c837d6f0c.2020.06.30
5A_YdRIAT3m1Z-vbqBuGWA 3 1 24 0 0 0
green open infra-000003 1-
goREK1QUKIQPAIVkWVaQ 3 1 226719 0 295 147
green open .security zeT65uOuRTKZMjg_bbUc1g
1 1 5 0 0 0
green open .kibana-377444158_kubeadmin wwMhDwJkR-
mRZQO84K0gUQ 3 1 1 0 0 0
green open infra-000006 5H-
KBSXGQKiO7hdapDE23g 3 1 226676 0 295 147
green open project.nw.6233ad57-aff0-4d5a-976f-370636f47b11.2020.06.30
dtc6J-nLSCC59EygeV41RQ 3 1 10 0 0 0
green open infra-000001 eH53BQ-
bSxSWR5xYZB6IVg 3 1 341800 0 443 220
green open .kibana-6
RVp7TemSSemGJcsSUmf3A 1 1 4 0 0 0
green open infra-000011
J7XWBauWSTe0jnzX02fU6A 3 1 226100 0 293 146
green open app-000001
axSAFfONQDmKwatKjPXdtw 3 1 103186 0 126 57
green open infra-000016
m9c1iRLtStWSF1GopaRyCg 3 1 13685 0 19 9
green open infra-000002 Hz6WvINtTvKcQzw-
ewmbYg 3 1 228994 0 296 148
green open project.qt.2c05acbd-bc12-4275-91ab-84d180b53505.2020.06.30
MUm3eFJjSPKQOJWoHskKqw 3 1 12262 0 14 7
green open infra-000013 KR9mMFUpQI-
jraYtanylGw 3 1 228166 0 298 148
green open audit-000001
eERqLdLmQOiQDFES1LBATQ 3 1 0 0 0 0

```

- e. 로그 수집기가 4.6으로 업데이트되었는지 확인합니다.

```
$ oc get ds fluentd -o json | grep fluentd-init
```

**fluentd-init** 컨테이너가 표시되어야 합니다.

```
"containerName": "fluentd-init"
```

- f. 로그 시각화 프로그램이 Kibana CRD를 사용하여 4.6으로 업데이트되었는지 확인합니다.

```
$ oc get kibana kibana -o json
```

**준비** 상태의 Kibana Pod가 표시되어야 합니다.

```
[
  {
    "clusterCondition": {
      "kibana-5fdd766ffd-nb2jj": [
        {
          "lastTransitionTime": "2020-06-30T14:11:07Z",
          "reason": "ContainerCreating",
          "status": "True",
          "type": ""
        },
        {
          "lastTransitionTime": "2020-06-30T14:11:07Z",
          "reason": "ContainerCreating",
          "status": "True",
          "type": ""
        }
      ]
    },
    "deployment": "kibana",
    "pods": {
      "failed": [],
      "notReady": []
    },
    "ready": [],
    "replicaSets": [
      "kibana-5fdd766ffd"
    ],
    "replicas": 1
  }
]
```

- g. Curator가 4.6으로 업데이트되었는지 확인합니다.

```
$ oc get cronjob -o name
```

```
cronjob.batch/curator
cronjob.batch/elasticsearch-delete-app
cronjob.batch/elasticsearch-delete-audit
cronjob.batch/elasticsearch-delete-infra
cronjob.batch/elasticsearch-rollover-app
cronjob.batch/elasticsearch-rollover-audit
cronjob.batch/elasticsearch-rollover-infra
```

**elasticsearch-delete-\*** 및 **elasticsearch-rollover-\*** 인덱스가 표시되어야 합니다.

## 업데이트 후 작업

Log Forwarding API를 사용하여 로그를 전달하는 경우 Elasticsearch Operator 및 클러스터 로깅 Operator가 4.6으로 완전히 업데이트된 후 **LogForwarding** 사용자 정의 리소스(CR)를 **ClusterLogForwarder** CR로 바꿔야 합니다.

## 7.2. 로그 전달 사용자 정의 리소스 업데이트

OpenShift Container Platform Log Forward API는 기술 프리뷰에서 OpenShift Container Platform 4.6에서 일반적으로 사용 가능으로 승격되었습니다. GA 릴리스에는 클러스터 로깅 사용자 정의 리소스(CR)를

변경하고 **LogForwarding** 사용자 정의 리소스(CR)를 **ClusterLogForwarder** CR로 교체해야 하는 몇 가지 개선 사항 및 향상된 사항이 포함되어 있습니다.

## OpenShift Container Platform 4.6의 샘플 ClusterLogForwarder 인스턴스

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: instance
  namespace: openshift-logging
....
spec:
  outputs:
  - url: http://remote.elasticsearch.com:9200
    name: elasticsearch
    type: elasticsearch
  - url: tls://fluentdserver.example.com:24224
    name: fluentd
    type: fluentdForward
    secret:
      name: fluentdserver
  pipelines:
  - inputRefs:
    - infrastructure
    - application
    name: mylogs
    outputRefs:
    - elasticsearch
  - inputRefs:
    - audit
    name: auditlogs
    outputRefs:
    - fluentd
    - default
  ...
```

## OpenShift Container Platform 4.5의 샘플 LogForwarding CR

```
apiVersion: logging.openshift.io/v1alpha1
kind: LogForwarding
metadata:
  name: instance
  namespace: openshift-logging
spec:
  disableDefaultForwarding: true
  outputs:
  - name: elasticsearch
    type: elasticsearch
    endpoint: remote.elasticsearch.com:9200
  - name: fluentd
    type: forward
    endpoint: fluentdserver.example.com:24224
    secret:
      name: fluentdserver
  pipelines:
```



```
- inputSource: logs.infra
  name: infra-logs
  outputRefs:
  - elasticsearch
- inputSource: logs.app
  name: app-logs
  outputRefs:
  - elasticsearch
- inputSource: logs.audit
  name: audit-logs
  outputRefs:
  - fluentd
```

다음 절차는 변경해야 하는 각 매개변수를 보여줍니다.

## 프로시저

4.5의 **LogForwarding** CR을 4.6용 **ClusterLogForwarding** CR로 업데이트하려면 다음과 같이 수정합니다.

1. 클러스터 로깅 사용자 정의 리소스(CR)를 편집하여 **logforwardingtechpreview** 주석을 제거합니다.

### 샘플 클러스터 로깅 CR

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  annotations:
    clusterlogging.openshift.io/logforwardingtechpreview: enabled ❶
  name: "instance"
  namespace: "openshift-logging"
....
```

- ❶ **logforwardingtechpreview** 주석을 제거합니다.

2. **LogForwarding** CR을 내보내 **ClusterLogForwarder** 인스턴스에 대한 YAML 파일을 만듭니다.

```
$ oc get LogForwarding instance -n openshift-logging -o yaml | tee ClusterLogForwarder.yaml
```

3. YAML 파일을 편집하여 다음과 같이 수정합니다.

### OpenShift Container Platform 4.6의 샘플 **ClusterLogForwarder** 인스턴스

```
apiVersion: logging.openshift.io/v1 ❶
kind: ClusterLogForwarder ❷
metadata:
  name: instance
  namespace: openshift-logging
....
spec: ❸
  outputs:
  - url: http://remote.elasticsearch.com:9200 ❹
```

```

name: elasticsearch
type: elasticsearch
- url: tls://fluentdserver.example.com:24224
name: fluentd
type: fluentdForward 5
secret:
  name: fluentdserver
pipelines:
- inputRefs: 6
  - infrastructure
  - application
name: mylogs
outputRefs:
  - elasticsearch
- inputRefs:
  - audit
name: auditlogs
outputRefs:
  - fluentd
  - default 7
...

```

- 1 **apiVersion**을 "logging.openshift.io/v1alpha1"에서 "logging.openshift.io/v1"로 변경합니다.
- 2 오브젝트 종류를 종류: "LogForwarding"에서 종류: "ClusterLogForwarder"로 변경합니다.
- 3 **disableDefaultForwarding: true** 매개변수를 제거합니다.
- 4 출력 매개변수를 **spec.outputs.endpoint**에서 **spec.outputs.url**로 변경합니다. 접두사가 없는 경우 **https://, tcp://** 등과 같은 접두사를 URL에 추가합니다.
- 5 Fluentd 출력의 경우 유형을 **forward**에서 **fluentdForward**로 변경합니다.
- 6 파이프라인 변경:
  - **spec.pipelines.inputSource**를 **spec.pipelines.inputRefs**로 변경
  - **logs.infra**를 인프라로 변경
  - **logs.app**을 애플리케이션으로 변경
  - **logs.audit**를 감사로 변경
- 7 선택사항: 기본 파이프라인을 추가하여 내부 Elasticsearch 인스턴스로 로그를 보냅니다. 기본 출력을 구성할 필요는 없습니다.



## 참고

내부 OpenShift Container Platform Elasticsearch 인스턴스에만 로그를 전달하려면 Log Forwarding API를 구성하지 마십시오.

4. CR 오브젝트를 생성합니다.

■

```
$ oc create -f ClusterLogForwarder.yaml
```

Log Forwarding API의 새로운 기능에 대한 자세한 내용은 [타사 시스템에 로그 전달](#)을 참조하십시오.

## 8장. 클러스터 대시보드 보기

OpenShift Container Platform 웹 콘솔의 **Logging / Elasticsearch** 노드 및 **Openshift Logging** 대시보드는 문제를 예방하고 진단하는 데 사용할 수 있는 Elasticsearch 인스턴스 및 개별 Elasticsearch 노드에 대한 심층적인 세부 정보를 보여줍니다.

**OpenShift 로깅** 대시보드에는 클러스터 리소스, 가비지 수집, 클러스터의 shard 및 Fluentd 통계를 포함하여 클러스터 수준에서 Elasticsearch 인스턴스에 대한 세부 정보를 보여주는 차트가 포함되어 있습니다.

**로깅/Elasticsearch 노드** 대시보드에는 인덱싱, shard, 리소스 등에 대한 세부 정보를 포함하여 노드 수준에서 많은 Elasticsearch 인스턴스에 대한 세부 정보를 보여주는 차트가 포함되어 있습니다.



### 참고

더 자세한 데이터를 보려면 대시보드에서 **Grafana UI** 링크를 클릭하여 Grafana 대시보드를 시작합니다. Grafana는 [OpenShift 클러스터 모니터링](#)과 함께 제공됩니다.

### 8.1. ELASTISEARCH 및 OPENSIFT 로깅 대시보드에 액세스

OpenShift Container Platform 웹 콘솔에서 **로깅/Elasticsearch 노드** 및 **Openshift 로깅** 대시보드를 볼 수 있습니다.

#### 프로시저

대시보드를 시작하려면 다음을 수행합니다.

1. OpenShift Container Platform 웹 콘솔에서 **모니터링** → **대시보드**를 클릭합니다.
2. **대시보드** 페이지의 **대시보드** 메뉴에서 **로깅/Elasticsearch 노드** 또는 **Openshift 로깅**을 선택합니다.  
**로깅/Elasticsearch 노드** 대시보드의 경우 보려는 Elasticsearch 노드를 선택하고 데이터 해상도를 설정할 수 있습니다.  
 여러 데이터 차트를 보여주는 적절한 대시보드가 표시됩니다.
3. 선택적으로 **시간 범위** 및 **새로 고침 간격** 메뉴에서 데이터를 표시하거나 새로 고칠 다른 시간 범위를 선택합니다.



### 참고

더 자세한 데이터를 보려면 **Grafana UI** 링크를 클릭하여 Grafana 대시보드를 시작합니다.

대시보드 차트에 대한 정보는 [OpenShift 로깅 대시보드 정보](#) 및 [로깅/Elasticsearch 노드 대시보드 정보](#)를 참조하십시오.

### 8.2. OPENSIFT 로깅 대시보드 정보

**OpenShift 로깅** 대시보드에는 문제를 진단하고 예측하는 데 사용할 수 있는 클러스터 수준에서 Elasticsearch 인스턴스에 대한 세부 정보를 보여주는 차트가 포함되어 있습니다.

#### 표 8.1. OpenShift 로깅 차트

지표	설명
Elastic 클러스터 상태	<p>현재 Elasticsearch 상태:</p> <ul style="list-style-type: none"> <li>● 온라인 - Elasticsearch 인스턴스가 온라인 상태를 나타냅니다.</li> <li>● 오프라인 - Elasticsearch 인스턴스가 오프라인 상태를 나타냅니다.</li> </ul>
Elastic 노드	Elasticsearch 인스턴스의 총 Elasticsearch 노드 수입니다.
Elastic Shard	Elasticsearch 인스턴스의 총 Elasticsearch shard 수입니다.
Elastic 문서	Elasticsearch 인스턴스의 총 Elasticsearch 문서 수입니다.
디스크의 총 인덱스 크기	Elasticsearch 인덱스에 사용 중인 총 디스크 공간입니다.
Elastic 보류 작업	인덱스 생성, 인덱스 매핑, shard 할당 또는 shard 오류와 같이 완료되지 않은 Elasticsearch 변경의 총 수입니다.
Elastic JVM GC 시간	JVM이 클러스터에서 Elasticsearch 가비지 수집 작업을 실행하는 데 소비한 시간입니다.
Elastic JVM GC 속도	JVM이 초당 가비지 활동을 실행한 총 횟수입니다.
Elastic 쿼리/가져오기 지연 시간 합계	<ul style="list-style-type: none"> <li>● 쿼리 지연 시간: 각 Elasticsearch 검색 쿼리를 실행하는 데 걸리는 평균 시간입니다.</li> <li>● 가져오기 지연 시간: 각 Elasticsearch 검색 쿼리가 데이터를 가져오는 데 소요되는 평균 시간입니다.</li> </ul> <p>가져오기 지연 시간은 일반적으로 쿼리 지연 시간보다 더 짧습니다. 가져오기 지연 시간이 지속적으로 증가하는 경우 느린 디스크, 데이터 보강 또는 결과가 너무 많은 대규모 요청을 나타낼 수 있습니다.</p>
Elastic 쿼리 속도	각 Elasticsearch 노드에 대해 Elasticsearch 인스턴스에 대해 실행된 초당 총 쿼리입니다.
CPU	Elasticsearch, Fluentd 및 Kibana에서 사용하는 CPU 양(각 구성 요소에 대해 표시됨).

지표	설명
사용된 Elastic JVM 힙	사용된 JVM 메모리 양입니다. 정상 클러스터에서 그래프는 JVM 가비지 수집에 의해 메모리가 해제됨에 따라 정기적으로 감소를 표시합니다.
Elasticsearch 디스크 사용량	각 Elasticsearch 노드에 대해 Elasticsearch 인스턴스에서 사용하는 총 디스크 공간입니다.
사용 중인 파일 설명자	Elasticsearch, Fluentd 및 Kibana에서 사용하는 총 파일 설명자 수입니다.
FluentD 방출 수	Fluentd 기본 출력에 대한 초당 총 Fluentd 메시지 수 및 기본 출력에 대한 재시도 횟수입니다.
FluentD 버퍼 가용성	체크에 사용할 수 있는 Fluentd 버퍼의 백분율입니다. 가득 찬 버퍼는 Fluentd가 수신된 로그 수를 처리할 수 없음을 나타낼 수 있습니다.
Elastic rx 바이트	Elasticsearch가 FluentD, Elasticsearch 노드 및 기타 소스에서 수신한 총 바이트 수입니다.
Elastic 인덱스 실패율	Elasticsearch 인덱스가 실패하는 초당 총 횟수입니다. 높은 비율은 인덱싱 문제를 나타낼 수 있습니다.
FluentD 출력 오류율	FluentD가 로그를 출력할 수 없는 초당 총 횟수입니다.

### 8.3. 로깅/ELASTICSEARCH 노드 대시보드의 차트

로깅/Elasticsearch 노드 대시보드에는 추가 진단을 위해 많은 노드 수준에서 Elasticsearch 인스턴스에 대한 세부 정보를 보여주는 차트가 포함되어 있습니다.

#### Elasticsearch 상태

로깅/Elasticsearch 노드 대시보드에는 Elasticsearch 인스턴스의 상태에 대한 다음 차트가 포함되어 있습니다.

표 8.2. Elasticsearch 상태 필드

지표	설명
----	----

지표	설명
클러스터 상태	<p>Elasticsearch 녹색, 노란색 및 빨간색 상태를 사용하여 선택한 기간 동안의 클러스터 상태:</p> <ul style="list-style-type: none"> <li>● 0 - Elasticsearch 인스턴스가 녹색 상태임을 나타냅니다. 이는 모든 shard가 할당되었음을 의미합니다.</li> <li>● 1 - Elasticsearch 인스턴스가 노란색 상태임을 나타냅니다. 이는 하나 이상의 shard에 대한 복제본 shard가 할당되지 않았음을 의미합니다.</li> <li>● 2 - Elasticsearch 인스턴스가 빨간색 상태임을 나타냅니다. 이는 하나 이상의 기본 shard와 해당 복제본이 할당되지 않았음을 의미합니다.</li> </ul>
클러스터 노드	클러스터의 총 Elasticsearch 노드 수입입니다.
클러스터 데이터 노드	클러스터에 있는 Elasticsearch 데이터 노드의 수입입니다.
클러스터 보류 작업	완료되지 않고 클러스터 큐에서 대기 중인 클러스터 상태 변경 수(예: 인덱스 생성, 인덱스 삭제 또는 shard 할당)입니다. 증가 추세는 클러스터가 변경 사항을 따라갈 수 없음을 나타냅니다.

### Elasticsearch 클러스터 인덱스 shard 상태

각 Elasticsearch 인덱스는 지속되는 데이터의 기본 단위인 하나 이상의 shard로 구성된 논리적 그룹입니다. 인덱스 shard는 기본 shard와 복제본 shard의 두 가지 유형이 있습니다. 문서가 인덱스로 인덱싱되면 기본 shard 중 하나에 저장되고 해당 shard의 모든 복제본에 복사됩니다. 기본 shard의 수는 인덱스가 생성될 때 지정되며 인덱스 수명 중에는 변경할 수 없습니다. 언제든지 복제본 shard 수를 변경할 수 있습니다.

인덱스 shard는 수명주기 단계 또는 클러스터에서 발생하는 이벤트에 따라 여러 상태가 될 수 있습니다. shard가 검색 및 인덱싱 요청을 수행할 수 있으면 shard가 활성화됩니다. shard가 이러한 요청을 수행할 수 없는 경우 shard는 비활성 상태입니다. shard가 초기화, 재할당, 할당 해제 등의 경우 shard는 비활성 상태일 수 있습니다.

인덱스 shard는 데이터의 물리적 표현인 인덱스 세그먼트라고 하는 여러 개의 작은 내부 블록으로 구성됩니다. 인덱스 세그먼트는 Lucene이 새로 인덱싱된 데이터를 커밋할 때 생성되는 비교적 작고 변경 불가능한 Lucene 인덱스입니다. Elasticsearch에서 사용하는 검색 라이브러리인 Lucene은 인덱스 세그먼트를 백그라운드에서 더 큰 세그먼트로 병합하여 총 세그먼트 수를 낮게 유지합니다. 세그먼트 병합 프로세스가 새 세그먼트가 생성되는 속도보다 느리면 문제가 있을 수 있습니다.

Lucene이 검색 작업과 같은 데이터 작업을 수행할 때 Lucene은 관련 인덱스의 인덱스 세그먼트에 대해 작업을 수행합니다. 이를 위해 각 세그먼트에는 메모리에 로드되고 매핑되는 특정 데이터 구조가 포함됩니다. 인덱스 매핑은 세그먼트 데이터 구조에서 사용하는 메모리에 상당한 영향을 미칠 수 있습니다.

로깅/Elasticsearch 노드 대시보드에는 Elasticsearch 인덱스 shard에 대한 다음 차트가 포함되어 있습니다.

표 8.3. Elasticsearch 클러스터 shard 상태 차트

지표	설명
클러스터 활성 shard	클러스터의 활성 기본 shard 수 및 복제본을 포함한 총 shard 수입니다. shard 수가 증가하면 클러스터 성능이 저하되기 시작할 수 있습니다.
클러스터 초기화 shard	클러스터의 비활성 shard 수입니다. 비활성 shard는 초기화 중이거나 다른 노드에 재 할당되거나 할당되지 않은 shard입니다. 일반적으로 클러스터에는 짧은 기간 동안 비활성 shard가 있습니다. 장기간에 걸쳐 비활성 shard 수가 증가하면 문제를 나타낼 수 있습니다.
클러스터 재배포 shard	Elasticsearch가 새 노드로 재배포하는 shard 수입니다. Elasticsearch는 노드의 높은 메모리 사용 또는 새 노드가 클러스터에 추가된 후와 같은 여러 가지 이유로 노드를 재배포합니다.
할당되지 않은 shard 클러스터	할당되지 않은 shard 수 Elasticsearch shard는 새 인덱스 추가 또는 노드 장애와 같은 이유로 할당 해제될 수 있습니다.

### Elasticsearch 노드 지표

각 Elasticsearch 노드에는 작업을 처리하는 데 사용할 수 있는 한정된 양의 리소스가 있습니다. 모든 리소스가 사용되고 Elasticsearch가 새 작업을 수행하려고 하면 Elasticsearch는 일부 리소스를 사용할 수 있을 때까지 작업을 큐에 넣습니다.

로깅/Elasticsearch 노드 대시보드에는 선택한 노드의 리소스 사용량과 Elasticsearch 큐에서 대기 중인 작업 수에 대한 다음 차트가 포함되어 있습니다.

표 8.4. Elasticsearch 노드 지표 차트

지표	설명
ThreadPool 작업	작업 유형별로 표시되는 개별 큐의 대기 작업 수입니다. 큐에 작업이 장기간 누적되면 노드 리소스 부족 또는 기타 문제가 있을 수 있습니다.
CPU 사용량	선택한 Elasticsearch 노드에서 사용 중인 CPU 양(호스트 컨테이너에 할당된 총 CPU의 백분율)입니다.
메모리 사용량	선택한 Elasticsearch 노드에서 사용 중인 메모리 양입니다.
디스크 사용량	선택한 Elasticsearch 노드에서 인덱스 데이터 및 메타 데이터에 사용되는 총 디스크 공간입니다.
문서 색인 비율	선택한 Elasticsearch 노드에서 문서가 인덱싱되는 비율입니다.



지표	설명
인덱싱 지연 시간	선택한 Elasticsearch 노드에서 문서를 인덱싱하는 데 걸린 시간입니다. 인덱싱 지연 시간은 JVM 힙 메모리 및 전체 로드와 같은 여러 요인의 영향을 받을 수 있습니다. 지연 시간이 증가하면 인스턴스의 리소스 용량이 부족함을 나타냅니다.
검색률	선택한 Elasticsearch 노드에서 실행되는 검색 요청 수입니다.
검색 지연 시간	선택한 Elasticsearch 노드에서 검색 요청을 완료하는 데 걸린 시간입니다. 검색 지연 시간은 여러 요인의 영향을 받을 수 있습니다. 지연 시간이 증가하면 인스턴스의 리소스 용량이 부족함을 나타냅니다.
문서 수(복제본 포함)	노드에 할당된 기본 shard와 복제본 shard 모두에 저장된 문서를 포함하여 선택한 Elasticsearch 노드에 저장된 Elasticsearch 문서 수입니다.
문서 삭제 비율	선택한 Elasticsearch 노드에 할당된 인덱스 shard에서 삭제되는 Elasticsearch 문서의 수입니다.
문서 병합 비율	선택한 Elasticsearch 노드에 할당된 인덱스 shard에서 병합되는 Elasticsearch 문서의 수입니다.

### Elasticsearch 노드 필드 데이터

**Fielddata**는 인덱스의 용어 목록을 보유하고 JVM 힙에 보관되는 Elasticsearch 데이터 구조입니다. 필드 데이터 구축은 비용이 많이 드는 작업이므로 Elasticsearch는 필드 데이터 구조를 캐시합니다. Elasticsearch는 기본 인덱스 세그먼트가 삭제 또는 병합되거나 모든 필드 데이터 캐시에 대한 JVM HEAP 메모리가 충분하지 않은 경우 필드 데이터 캐시를 제거할 수 있습니다.

로깅/Elasticsearch 노드 대시보드에는 Elasticsearch 필드 데이터에 대한 다음 차트가 포함되어 있습니다.

표 8.5. Elasticsearch 노드 필드 데이터 차트

지표	설명
Fielddata 메모리 크기	선택한 Elasticsearch 노드에서 필드 데이터 캐시에 사용된 JVM 힙의 양입니다.
Fielddata 제거	선택한 Elasticsearch 노드에서 삭제된 fielddata 구조의 수입니다.

### Elasticsearch 노드 쿼리 캐시

인덱스에 저장된 데이터가 변경되지 않으면 Elasticsearch에서 재사용할 수 있도록 검색 쿼리 결과가 노드 수준 쿼리 캐시에 캐시됩니다.

로깅/Elasticsearch 노드 대시보드에는 Elasticsearch 노드 쿼리 캐시에 대한 다음 차트가 포함되어 있습니다.

표 8.6. Elasticsearch 노드 쿼리 차트

지표	설명
쿼리 캐시 크기	선택한 Elasticsearch 노드에 할당된 모든 shard의 쿼리 캐시에 사용된 총 메모리 양입니다.
쿼리 캐시 제거	선택한 Elasticsearch 노드의 쿼리 캐시 제거 수입입니다.
쿼리 캐시 적중	선택한 Elasticsearch 노드의 쿼리 캐시 적중 수입입니다.
쿼리 캐시 누락	선택한 Elasticsearch 노드의 쿼리 캐시 누락 수입입니다.

### Elasticsearch 인덱스 제한

문서를 인덱싱할 때 Elasticsearch는 데이터의 물리적 표현인 인덱스 세그먼트에 문서를 저장합니다. 동시에 Elasticsearch는 리소스 사용을 최적화하기 위해 주기적으로 작은 세그먼트를 큰 세그먼트로 병합합니다. 인덱싱이 세그먼트 병합 기능보다 빠르면 병합 프로세스가 충분히 빨리 완료되지 않아 검색 및 성능에 문제가 발생할 수 있습니다. 이러한 상황을 방지하기 위해 Elasticsearch는 일반적으로 인덱싱에 할당된 스레드 수를 단일 스레드로 줄여 인덱싱을 제한합니다.

로깅/Elasticsearch 노드 대시보드에는 Elasticsearch 인덱스 조절에 대한 다음 차트가 포함되어 있습니다.

표 8.7. 인덱스 제한 차트

지표	설명
인덱싱 제한	Elasticsearch가 선택한 Elasticsearch 노드에서 인덱싱 작업을 제한한 시간입니다.
제한 병합	Elasticsearch가 선택한 Elasticsearch 노드에서 세그먼트 병합 작업을 제한한 시간입니다.

### 노드 JVM 힙 통계

로깅/Elasticsearch 노드 대시보드에는 JVM 힙 작업에 대한 다음 차트가 포함되어 있습니다.

표 8.8. JVM 힙 통계 차트

지표	설명
사용된 힙	선택한 Elasticsearch 노드에서 사용되는 총 할당된 JVM 힙 공간의 양입니다.
GC 수	오래된 가비지 수집에 의해 선택된 Elasticsearch 노드에서 실행된 가비지 수집 작업의 수입입니다.

지표	설명
GC 시간	JVM이 선택한 Elasticsearch 노드에서 가비지 수집 작업을 실행하는 데 소비한 시간(오래되고 오래되지 않은 가비지 수집 기준)입니다.

## 9장. 클러스터 로깅 문제 해결

### 9.1. 클러스터 로깅 상태 보기

클러스터 로깅 Operator 및 여러 클러스터 로깅 구성 요소의 상태를 볼 수 있습니다.

#### 9.1.1. 클러스터 로깅 Operator의 상태 보기

클러스터 로깅 Operator의 상태를 볼 수 있습니다.

##### 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

##### 프로시저

1. **openshift-logging** 프로젝트로 변경합니다.

```
$ oc project openshift-logging
```

2. 클러스터 로깅 상태를 보려면 다음을 수행합니다.

- a. 클러스터 로깅 상태를 가져옵니다.

```
$ oc get clusterlogging instance -o yaml
```

##### 출력 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging

....

status: ❶
  collection:
    logs:
      fluentdStatus:
        daemonSet: fluentd ❷
        nodes:
          fluentd-2rhqp: ip-10-0-169-13.ec2.internal
          fluentd-6fgjh: ip-10-0-165-244.ec2.internal
          fluentd-6l2ff: ip-10-0-128-218.ec2.internal
          fluentd-54nx5: ip-10-0-139-30.ec2.internal
          fluentd-flpnn: ip-10-0-147-228.ec2.internal
          fluentd-n2frh: ip-10-0-157-45.ec2.internal
        pods:
          failed: []
          notReady: []
          ready:
            - fluentd-2rhqp
            - fluentd-54nx5
            - fluentd-6fgjh
            - fluentd-6l2ff
```

```

- fluentd-flpnn
- fluentd-n2frh
logstore: ❸
elasticsearchStatus:
- ShardAllocationEnabled: all
cluster:
  activePrimaryShards: 5
  activeShards: 5
  initializingShards: 0
  numDataNodes: 1
  numNodes: 1
  pendingTasks: 0
  relocatingShards: 0
  status: green
  unassignedShards: 0
clusterName: elasticsearch
nodeConditions:
  elasticsearch-cdm-mkkdys93-1:
nodeCount: 1
pods:
  client:
    failed:
    notReady:
    ready:
    - elasticsearch-cdm-mkkdys93-1-7f7c6-mjm7c
  data:
    failed:
    notReady:
    ready:
    - elasticsearch-cdm-mkkdys93-1-7f7c6-mjm7c
  master:
    failed:
    notReady:
    ready:
    - elasticsearch-cdm-mkkdys93-1-7f7c6-mjm7c
visualization: ❹
kibanaStatus:
- deployment: kibana
pods:
  failed: []
  notReady: []
  ready:
  - kibana-7fb4fd4cc9-f2nls
replicaSets:
- kibana-7fb4fd4cc9
replicas: 1

```

- ❶ 출력에서 클러스터 상태 필드가 **상태** 스탠자에 나타납니다.
- ❷ Fluentd 포트에 대한 정보.
- ❸ Elasticsearch 클러스터 건강, 녹색, 노란색 또는 빨간색을 포함한 Elasticsearch 포트에 대한 정보입니다.
- ❹ Kibana 포트에 대한 정보.

### 9.1.1.1. 조건 메시지 예

다음은 클러스터 로깅 인스턴스의 **Status.Nodes** 섹션에 있는 일부 조건 메시지의 예입니다.

다음과 유사한 상태 메시지는 노드가 구성된 낮은 워터마크를 초과했으며 이 노드에 shard가 할당되지 않았음을 나타냅니다.

#### 출력 예

```
nodes:
- conditions:
- lastTransitionTime: 2019-03-15T15:57:22Z
  message: Disk storage usage for node is 27.5gb (36.74%). Shards will be not
    be allocated on this node.
  reason: Disk Watermark Low
  status: "True"
  type: NodeStorage
  deploymentName: example-elasticsearch-clientdatamaster-0-1
  upgradeStatus: {}
```

다음과 유사한 상태 메시지는 노드가 구성된 상위 워터마크를 초과했으며 shard가 다른 노드로 재배치됨을 나타냅니다.

#### 출력 예

```
nodes:
- conditions:
- lastTransitionTime: 2019-03-15T16:04:45Z
  message: Disk storage usage for node is 27.5gb (36.74%). Shards will be relocated
    from this node.
  reason: Disk Watermark High
  status: "True"
  type: NodeStorage
  deploymentName: cluster-logging-operator
  upgradeStatus: {}
```

다음과 유사한 상태 메시지는 CR의 Elasticsearch 노드 선택기가 클러스터의 노드와 일치하지 않음을 나타냅니다.

#### 출력 예

```
Elasticsearch Status:
Shard Allocation Enabled: shard allocation unknown
Cluster:
  Active Primary Shards: 0
  Active Shards: 0
  Initializing Shards: 0
  Num Data Nodes: 0
  Num Nodes: 0
  Pending Tasks: 0
  Relocating Shards: 0
  Status: cluster health unknown
  Unassigned Shards: 0
Cluster Name: elasticsearch
Node Conditions:
```

```

elasticsearch-cdm-mkkdys93-1:
Last Transition Time: 2019-06-26T03:37:32Z
Message:          0/5 nodes are available: 5 node(s) didn't match node selector.
Reason:           Unschedulable
Status:           True
Type:             Unschedulable
elasticsearch-cdm-mkkdys93-2:
Node Count: 2
Pods:
Client:
Failed:
Not Ready:
  elasticsearch-cdm-mkkdys93-1-75dd69dccc-f7f49
  elasticsearch-cdm-mkkdys93-2-67c64f5f4c-n58vl
Ready:
Data:
Failed:
Not Ready:
  elasticsearch-cdm-mkkdys93-1-75dd69dccc-f7f49
  elasticsearch-cdm-mkkdys93-2-67c64f5f4c-n58vl
Ready:
Master:
Failed:
Not Ready:
  elasticsearch-cdm-mkkdys93-1-75dd69dccc-f7f49
  elasticsearch-cdm-mkkdys93-2-67c64f5f4c-n58vl
Ready:

```

다음과 유사한 상태 메시지는 요청한 PVC가 PV에 바인딩할 수 없음을 나타냅니다.

#### 출력 예

```

Node Conditions:
elasticsearch-cdm-mkkdys93-1:
Last Transition Time: 2019-06-26T03:37:32Z
Message:          pod has unbound immediate PersistentVolumeClaims (repeated 5 times)
Reason:           Unschedulable
Status:           True
Type:             Unschedulable

```

다음과 유사한 상태 메시지는 노드 선택기가 노드와 일치하지 않기 때문에 Fluentd 포드를 예약할 수 없음을 나타냅니다.

#### 출력 예

```

Status:
Collection:
Logs:
Fluentd Status:
Daemon Set: fluentd
Nodes:
Pods:
Failed:
Not Ready:
Ready:

```

### 9.1.2. 클러스터 로깅 구성 요소의 상태 보기

여러 클러스터 로깅 구성 요소의 상태를 볼 수 있습니다.

#### 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

#### 프로시저

- openshift-logging** 프로젝트로 변경합니다.

```
$ oc project openshift-logging
```

- 클러스터 로깅 배포의 상태 보기:

```
$ oc describe deployment cluster-logging-operator
```

#### 출력 예

```
Name:          cluster-logging-operator
....

Conditions:
  Type           Status Reason
  ----           -
  Available      True   MinimumReplicasAvailable
  Progressing    True   NewReplicaSetAvailable
....

Events:
  Type    Reason           Age    From           Message
  ----    -
  Normal  ScalingReplicaSet 62m    deployment-controller Scaled up replica set cluster-logging-operator-574b8987df to 1----
```

- 클러스터 로깅 ReplicaSet 보기:
  - ReplicaSet의 이름을 가져옵니다.

#### 출력 예

```
$ oc get replicaset
```

#### 출력 예

NAME	DESIRED	CURRENT	READY	AGE
cluster-logging-operator-574b8987df	1	1	1	159m
elasticsearch-cdm-uhr537yu-1-6869694fb	1	1	1	157m
elasticsearch-cdm-uhr537yu-2-857b6d676f	1	1	1	156m
elasticsearch-cdm-uhr537yu-3-5b6fdd8cfd	1	1	1	155m
kibana-5bd5544f87	1	1	1	157m



- 
- b. ReplicaSet의 상태를 가져옵니다.

```
$ oc describe replicaset cluster-logging-operator-574b8987df
```

#### 출력 예

```
Name:          cluster-logging-operator-574b8987df
....

Replicas:      1 current / 1 desired
Pods Status:   1 Running / 0 Waiting / 0 Succeeded / 0 Failed
....

Events:
  Type    Reason             Age   From              Message
  ----    -
  Normal  SuccessfulCreate   66m   replicaset-controller Created pod: cluster-logging-operator-574b8987df-qjhqv----
```

## 9.2. 로그 저장소의 상태 보기

Elasticsearch Operator 및 여러 Elasticsearch 구성 요소의 상태를 볼 수 있습니다.

### 9.2.1. 로그 저장소의 상태 보기

로그 저장소의 상태를 볼 수 있습니다.

#### 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

#### 프로시저

1. **openshift-logging** 프로젝트로 변경합니다.

```
$ oc project openshift-logging
```

2. 상태를 보려면 다음을 수행합니다.

- a. 로그 저장소 인스턴스의 이름을 가져옵니다.

```
$ oc get Elasticsearch
```

#### 출력 예

```
NAME          AGE
elasticsearch  5h9m
```

- b. 로그 저장소 상태를 가져옵니다.

```
$ oc get Elasticsearch <Elasticsearch-instance> -o yaml
```

예를 들면 다음과 같습니다.

```
$ oc get Elasticsearch elasticsearch -n openshift-logging -o yaml
```

출력에는 다음과 유사한 정보가 포함됩니다.

#### 출력 예

```
status: 1
cluster: 2
  activePrimaryShards: 30
  activeShards: 60
  initializingShards: 0
  numDataNodes: 3
  numNodes: 3
  pendingTasks: 0
  relocatingShards: 0
  status: green
  unassignedShards: 0
clusterHealth: ""
conditions: [] 3
nodes: 4
- deploymentName: elasticsearch-cdm-zjf34ved-1
  upgradeStatus: {}
- deploymentName: elasticsearch-cdm-zjf34ved-2
  upgradeStatus: {}
- deploymentName: elasticsearch-cdm-zjf34ved-3
  upgradeStatus: {}
pods: 5
  client:
    failed: []
    notReady: []
    ready:
      - elasticsearch-cdm-zjf34ved-1-6d7fbf844f-sn422
      - elasticsearch-cdm-zjf34ved-2-dfbd988bc-qkzjz
      - elasticsearch-cdm-zjf34ved-3-c8f566f7c-t7zkt
  data:
    failed: []
    notReady: []
    ready:
      - elasticsearch-cdm-zjf34ved-1-6d7fbf844f-sn422
      - elasticsearch-cdm-zjf34ved-2-dfbd988bc-qkzjz
      - elasticsearch-cdm-zjf34ved-3-c8f566f7c-t7zkt
  master:
    failed: []
    notReady: []
    ready:
      - elasticsearch-cdm-zjf34ved-1-6d7fbf844f-sn422
      - elasticsearch-cdm-zjf34ved-2-dfbd988bc-qkzjz
      - elasticsearch-cdm-zjf34ved-3-c8f566f7c-t7zkt
shardAllocationEnabled: all
```

- 1 출력에서 클러스터 상태 필드가 **상태** 스탠자에 나타납니다.
- 2 로그 저장소의 상태:
  - 활성 기본 shard 수입니다.
  - 활성 shard 수입니다.
  - 초기화 중인 shard 수입니다.
  - 로그 저장소 데이터 노드 수입니다.
  - 총 로그 저장소 노드 수입니다.
  - 보류 중인 작업 수입니다.
  - 로그 저장소 상태는 **녹색, 빨간색, 노란색**입니다.
  - 할당되지 않은 shard 수
- 3 존재하는 경우 모든 상태 조건. 로그 저장소 상태는 포드를 배치할 수없는 경우 스케줄러의 사유를 나타냅니다. 다음 조건과 관련된 모든 이벤트가 표시됩니다.
  - 컨테이너 로그 저장소 및 프록시 컨테이너를 기다리는 중입니다.
  - 컨테이너 로그 저장소 및 프록시 컨테이너 모두에 대해 종료되었습니다.
  - 포드 예약 불가. 또한 여러 가지 문제에 대한 조건이 표시됩니다(**조건 메시지 예** 참조).
- 4 **upgradeStatus**가 있는 클러스터의 로그 저장소 노드.
- 5 'failed', **notReady** 또는 **준비** 상태 아래에 나열된 클러스터의 로그를 저장 클라이언트, 데이터 및 마스터 포드.

### 9.2.1.1. 조건 메시지 예

다음은 Elasticsearch 인스턴스의 **상태** 섹션에 있는 일부 조건 메시지의 예입니다.

이 상태 메시지는 노드가 구성된 낮은 워터마크를 초과했으며 노드에 shard가 할당되지 않았음을 나타냅니다.

```
status:
  nodes:
    - conditions:
      - lastTransitionTime: 2019-03-15T15:57:22Z
        message: Disk storage usage for node is 27.5gb (36.74%). Shards will be not
          be allocated on this node.
        reason: Disk Watermark Low
        status: "True"
        type: NodeStorage
      deploymentName: example-elasticsearch-cdm-0-1
      upgradeStatus: {}
```

이 상태 메시지는 노드가 구성된 상위 워터마크를 초과했으며 shard가 다른 노드로 재배치됨을 나타냅니다.

```

status:
  nodes:
  - conditions:
    - lastTransitionTime: 2019-03-15T16:04:45Z
      message: Disk storage usage for node is 27.5gb (36.74%). Shards will be relocated
        from this node.
      reason: Disk Watermark High
      status: "True"
      type: NodeStorage
    deploymentName: example-elasticsearch-cdm-0-1
    upgradeStatus: {}

```

이 상태 메시지는 CR의 로그 저장소 노드 선택기가 클러스터의 노드와 일치하지 않음을 나타냅니다.

```

status:
  nodes:
  - conditions:
    - lastTransitionTime: 2019-04-10T02:26:24Z
      message: '0/8 nodes are available: 8 node(s) didn't match node selector.'
      reason: Unschedulable
      status: "True"
      type: Unschedulable

```

이 상태 메시지는 로그 저장소 CR이 존재하지 않는 PVC를 사용함을 나타냅니다.

```

status:
  nodes:
  - conditions:
    - last Transition Time: 2019-04-10T05:55:51Z
      message: pod has unbound immediate PersistentVolumeClaims (repeated 5 times)
      reason: Unschedulable
      status: True
      type: Unschedulable

```

이 상태 메시지는 로그 저장소 클러스터에 로그 저장소 중복 정책을 지원하기에 충분한 노드가 없음을 나타냅니다.

```

status:
  clusterHealth: ""
  conditions:
  - lastTransitionTime: 2019-04-17T20:01:31Z
    message: Wrong RedundancyPolicy selected. Choose different RedundancyPolicy or
      add more nodes with data roles
    reason: Invalid Settings
    status: "True"
    type: InvalidRedundancy

```

이 상태 메시지는 클러스터에 마스터 노드가 너무 많음을 나타냅니다.

```

status:
  clusterHealth: green
  conditions:
  - lastTransitionTime: '2019-04-17T20:12:34Z'
    message: >-

```

```
Invalid master nodes count. Please ensure there are no more than 3 total
nodes with master roles
reason: Invalid Settings
status: 'True'
type: InvalidMasters
```

### 9.2.2. 로그 저장소 구성 요소의 상태 보기

여러 로그 저장소 구성 요소의 상태를 볼 수 있습니다.

#### Elasticsearch 인덱스

Elasticsearch 인덱스의 상태를 볼 수 있습니다.

1. Elasticsearch 포드의 이름을 가져옵니다.

```
$ oc get pods --selector component=elasticsearch -o name
```

#### 출력 예

```
pod/elasticsearch-cdm-1godmszn-1-6f8495-vp4lw
pod/elasticsearch-cdm-1godmszn-2-5769cf-9ms2n
pod/elasticsearch-cdm-1godmszn-3-f66f7d-zqkz7
```

2. 인덱스의 상태를 가져옵니다.

```
$ oc exec elasticsearch-cdm-4vjor49p-2-6d4d7db474-q2w7z -- indices
```

#### 출력 예

```
Defaulting container name to elasticsearch.
Use 'oc describe pod/elasticsearch-cdm-4vjor49p-2-6d4d7db474-q2w7z -n openshift-logging' to see all of the containers in this pod.

green open infra-000002                                S4QANnf1QP6NgCegfnrnbQ
3 1 119926 0 157 78
green open audit-000001                                8_EQx77iQCSTzFOXtxRqFw
3 1 0 0 0 0
green open .security                                   iDjscH7aSUGhldq0LheLBQ 1
1 5 0 0 0
green open .kibana_-377444158_kubeadmin                 yBywZ9GfSrKebz5gWBZbjw 3 1 1 0 0 0
green open infra-000001                                z6Dpe__ORgiopEpW6YI44A
3 1 871000 0 874 436
green open app-000001                                   hlrazQCeSlSewG3c2VlvsQ
3 1 2453 0 3 1
green open .kibana_1                                    JCitcBMSQxKOvIq6iQW6wg
1 1 0 0 0 0
green open .kibana_-1595131456_user1                   glYFIEGRRe-
ka0W3okS-mQ 3 1 1 0 0 0
```

#### 로그 저장소 포트

로그 저장소를 호스팅하는 포트의 상태를 볼 수 있습니다.

```
1. - - - - -
```

1. 포트 이름을 가져옵니다.

```
$ oc get pods --selector component=elasticsearch -o name
```

#### 출력 예

```
pod/elasticsearch-cdm-1godmszn-1-6f8495-vp4lw
pod/elasticsearch-cdm-1godmszn-2-5769cf-9ms2n
pod/elasticsearch-cdm-1godmszn-3-f66f7d-zqkz7
```

2. 포트 상태를 가져옵니다.

```
oc describe pod elasticsearch-cdm-1godmszn-1-6f8495-vp4lw
```

출력에는 다음 상태 정보가 포함됩니다.

#### 출력 예

```
....
Status:      Running

....

Containers:
  elasticsearch:
    Container ID:  cri-o://b7d44e0a9ea486e27f47763f5bb4c39dfd2
    State:        Running
      Started:    Mon, 08 Jun 2020 10:17:56 -0400
    Ready:        True
    Restart Count: 0
    Readiness:    exec [/usr/share/elasticsearch/probe/readiness.sh] delay=10s timeout=30s
                  period=5s #success=1 #failure=3

....

  proxy:
    Container ID:  cri-
o://3f77032abaddbb1652c116278652908dc01860320b8a4e741d06894b2f8f9aa1
    State:        Running
      Started:    Mon, 08 Jun 2020 10:18:38 -0400
    Ready:        True
    Restart Count: 0

....

Conditions:
  Type             Status
  Initialized      True
  Ready            True
  ContainersReady  True
  PodScheduled     True

....

Events:            <none>
```

## 로그 저장소 포드 배포 구성

로그 저장소 배포 구성의 상태를 볼 수 있습니다.

1. 배포 구성의 이름을 가져옵니다.

```
$ oc get deployment --selector component=elasticsearch -o name
```

### 출력 예

```
deployment.extensions/elasticsearch-cdm-1gon-1
deployment.extensions/elasticsearch-cdm-1gon-2
deployment.extensions/elasticsearch-cdm-1gon-3
```

2. 배포 구성 상태를 가져옵니다.

```
$ oc describe deployment elasticsearch-cdm-1gon-1
```

출력에는 다음 상태 정보가 포함됩니다.

### 출력 예

```
....
Containers:
  elasticsearch:
    Image: registry.redhat.io/openshift4/ose-logging-elasticsearch5:v4.3
    Readiness: exec [/usr/share/elasticsearch/probe/readiness.sh] delay=10s timeout=30s
               period=5s #success=1 #failure=3
....

Conditions:
  Type           Status  Reason
  ----           -
  Progressing    Unknown DeploymentPaused
  Available      True    MinimumReplicasAvailable
....

Events:          <none>
```

## 로그 저장소 ReplicaSet

로그 저장소 ReplicaSet의 상태를 볼 수 있습니다.

1. 복제 세트의 이름을 가져옵니다.

```
$ oc get replicaSet --selector component=elasticsearch -o name
```

```
replicaset.extensions/elasticsearch-cdm-1gon-1-6f8495
replicaset.extensions/elasticsearch-cdm-1gon-2-5769cf
replicaset.extensions/elasticsearch-cdm-1gon-3-f66f7d
```

2. 복제 세트의 상태를 가져옵니다.

```
$ oc describe replicaSet elasticsearch-cdm-1gon-1-6f8495
```

출력에는 다음 상태 정보가 포함됩니다.

#### 출력 예

```
....
Containers:
  elasticsearch:
    Image: registry.redhat.io/openshift4/ose-logging-
    elasticsearch6@sha256:4265742c7cdd85359140e2d7d703e4311b6497eec7676957f455d6
    908e7b1c25
    Readiness: exec [/usr/share/elasticsearch/probe/readiness.sh] delay=10s timeout=30s
    period=5s #success=1 #failure=3
....

Events:      <none>
```

## 9.3. 클러스터 로깅 경고 이해

모든 로깅 수집기 경고는 OpenShift Container Platform 웹 콘솔의 경고 UI에 나열됩니다.

### 9.3.1. 로깅 수집기 경고 보기

경고는 알림 UI의 **경고** 탭에서 OpenShift Container Platform 웹 콘솔에 표시됩니다. 경고는 다음 상태 중 하나입니다.

- **실행**. 시간 초과 기간 동안 경고 조건이 적용됩니다. 더 많은 정보를 보거나 경고를 끄려면 발사 경고의 끝에 있는 **옵션** 메뉴를 클릭합니다.
- **보류 중** 경고 조건이 현재 true이지만 시간 초과에 도달하지 않았습니다.
- **실행하지 않음**. 경고가 현재 트리거되지 않았습니다.

#### 프로시저

클러스터 로깅 및 기타 OpenShift Container Platform 경고를 보려면 다음을 수행합니다.

1. OpenShift Container Platform 콘솔에서 **모니터링** → **경고**를 클릭합니다.
2. **경고** 탭을 클릭합니다. 선택한 필터에 따라 경고가 나열됩니다.

#### 추가 리소스

- 경고 UI에 대한 자세한 내용은 [경고 관리](#)를 참조하십시오.

### 9.3.2. 로깅 수집기 경고 정보

로깅 수집기가 다음 경고를 생성합니다. 경고 UI의 **경고** 페이지의 OpenShift Container Platform 웹 콘솔에서 이 경고를 볼 수 있습니다.

#### 표 9.1. Fluentd Prometheus 경고



경고	메시지	설명	심각도
<b>FluentDHighErrorRate</b>	<b>fluentd &lt;instance&gt;에 의해 레코드의 &lt;value&gt;에서 오류가 발생했습니다.</b>	FluentD 출력 오류의 수는 높으며 기본적으로 이전 15분 동안 10개 이상입니다.	경고
<b>FluentdNodeDown</b>	<b>Prometheus는 fluentd &lt;인스턴스&gt;를 10분 이상 스크랩할 수 없습니다.</b>	Fluentd는 Prometheus가 특정 Fluentd 인스턴스를 스크랩할 수 없다고 보고했습니다.	심각
<b>FluentdQueueLengthBurst</b>	<b>마지막 순간에 fluentd &lt;instance&gt; 버퍼 큐 길이는 32보다 증가했습니다. 현재 값은 &lt;value&gt;입니다.</b>	Fluentd는 인덱싱되는 데이터를 유지할 수 없다고 보고합니다.	경고
<b>FluentdQueueLengthIncreasing</b>	<b>지난 12시간 동안 fluentd &lt;instance&gt; 버퍼 큐 길이는 1보다 지속적으로 증가했습니다. 현재 값은 &lt;value&gt;입니다.</b>	Fluentd는 큐 크기가 증가하고 있다고 보고합니다.	심각
<b>FluentDVeryHighErrorRate</b>	<b>fluentd &lt;instance&gt;에 의해 레코드의 &lt;value&gt;에서 오류가 발생했습니다.</b>	FluentD 출력 오류의 수는 기본적으로 이전 15분 동안 25개 이상으로 매우 높습니다.	심각

### 9.3.3. Elasticsearch 경고 규칙 정보

이러한 경고 규칙을 Prometheus에서 볼 수 있습니다.

경고	설명	심각도
ElasticsearchClusterNotHealthy	클러스터 상태가 2분 이상 빨간색이었습니다. 클러스터가 쓰기를 허용하지 않거나 shard가 누락되었거나 마스터 노드가 아직 선택되지 않았을 수 있습니다.	심각
ElasticsearchClusterNotHealthy	클러스터 상태가 최소 20분 동안 노란색이었습니다. 일부 shard 복제본이 할당되지 않았했습니다.	경고
ElasticsearchDiskSpaceRunningLow	클러스터는 향후 6시간 내에 디스크 공간이 부족할 것으로 예상됩니다.	심각
ElasticsearchHighFileDescriptorUsage	클러스터는 다음 시간 내에 파일 설명자가 없을 것으로 예상됩니다.	경고

경고	설명	심각도
ElasticsearchJVMHeapUseHigh	지정된 노드의 JVM 힙 사용량이 높습니다.	경고
ElasticsearchNodeDiskWatermarkReached	디스크 여유 공간이 부족하여 지정된 노드가 낮은 워터마크에 도달했습니다. 더 이상 shard를 이 노드에 할당할 수 없습니다. 노드에 디스크 공간을 추가하는 것을 고려해야 합니다.	정보
ElasticsearchNodeDiskWatermarkReached	디스크 여유 공간이 부족하여 지정된 노드가 높은 워터마크에 도달했습니다. 일부 shard는 가능한 경우 다른 노드에 다시 할당됩니다. 더 많은 디스크 공간이 노드에 추가되거나 이 노드에 할당된 오래된 인덱스를 삭제하십시오.	경고
ElasticsearchNodeDiskWatermarkReached	디스크 여유 공간이 부족하여 지정된 노드가 플러드 워터마크에 도달했습니다. 이 노드에 할당된 shard가 있는 모든 인덱스에는 읽기 전용 블록이 적용됩니다. 디스크 사용량이 높은 워터마크 아래로 떨어지면 인덱스 블록을 수동으로 해제해야 합니다.	심각
ElasticsearchJVMHeapUseHigh	지정된 노드의 JVM 힙 사용량이 너무 높습니다.	경고
ElasticsearchWriteRequestsRejectionJumps	Elasticsearch는 지정된 노드에서 쓰기 거부가 증가하고 있습니다. 이 노드는 인덱싱 속도를 따라가지 못할 수 있습니다.	경고
AggregatedLoggingSystemCPUHigh	지정된 노드의 시스템에서 사용하는 CPU가 너무 높습니다.	경고
ElasticsearchProcessCPUHigh	지정된 노드에서 Elasticsearch가 사용하는 CPU가 너무 높습니다.	경고

## 9.4. 로그 시각화 프로그램 문제 해결

OpenShift Container Platform과 함께 Kibana 콘솔을 사용하면 쉽게 해결할 수 있지만 유용한 오류 메시지가 표시되지 않는 문제가 발생할 수 있습니다. OpenShift Container Platform에 Kibana를 배포할 때 문제가 발생하면 다음 문제 해결 섹션을 확인하십시오.

### 9.4.1. Kibana 콘솔을 볼 때 Kubernetes 암호 오류 문제 해결

Kibana 콘솔을 방문하려고 하면 대신 브라우저 오류가 발생할 수 있습니다.

```
{"error": "invalid_request", "error_description": "The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed."}
```

OAuth2 클라이언트와 서버가 일치하지 않기 때문일 수 있습니다. 클라이언트의 반환 주소는 허용 목록에 있어야 로그인 후 서버가 안전하게 다시 리디렉션될 수 있습니다.

OAuthClient 항목을 교체하여 이 문제를 해결하십시오.

## 프로시저

OAuthClient 항목을 바꾸려면 다음을 수행합니다.

1. 다음 명령을 실행하여 현재 OAuthClient를 삭제합니다.

```
$ oc delete oauthclient/kibana-proxy
```

문제가 지속되면 OAuth 클라이언트에 나열된 URL에서 Kibana에 액세스하고 있는지 확인하십시오. 이 문제는 표준 443 HTTPS 포트 대신 1443과 같은 전달 포트에서 URL에 액세스하면 발생할 수 있습니다. OAuth 클라이언트를 편집하여 서버 허용 목록을 조정할 수 있습니다.

```
$ oc edit oauthclient/kibana-proxy
```

### 9.4.2. Kibana 콘솔을 볼 때 Kubernetes 503 오류 문제 해결

Kibana 콘솔을 볼 때 프록시 오류가 발생하면 다음 두 가지 문제 중 하나로 인해 발생할 수 있습니다.

- Kibana가 포드를 인식하지 못할 수 있습니다. Elasticsearch 시작 속도가 느리면 Kibana가 도달하는 데 시간이 초과될 수 있습니다. 관련 서비스에 엔드 포인트가 있는지 확인하십시오.

```
$ oc describe service kibana
```

#### 출력 예

```
Name:          kibana
[...]
Endpoints:     <none>
```

Kibana 포드가 활성화되어 있으면 엔드 포인트가 나열됩니다. 그렇지 않은 경우 Kibana 포드 및 배포 상태를 확인하십시오. 배포를 축소 후 다시 백업해야 할 수도 있습니다.

- Kibana 서비스에 액세스하기 위한 경로가 숨겨져 있습니다. 하나의 프로젝트에서 테스트 배포를 수행한 다음 첫 번째 배포를 완전히 제거하지 않고 다른 프로젝트에 배포할 경우에 발생할 수 있습니다. 여러 경로가 동일한 대상으로 전송되면 기본 라우터는 처음 작성된 경로로만 라우팅됩니다. 문제가있는 경로가 여러 위치에 정의되어 있는지 확인하십시오.

```
$ oc get route --all-namespaces --selector logging-infra=support
```

## 9.5. 로그 CURATOR 문제 해결

이 섹션의 정보를 사용하여 로그 큐를 디버깅할 수 있습니다. Curator는 OpenShift Container Platform 4.6 이전의 Elasticsearch 인덱스 형식의 데이터를 삭제하는 데 사용되며 이후 릴리스에서 제거됩니다.

### 9.5.1. 로그 큐레이션 문제 해결

이 섹션의 정보를 사용하여 로그 큐를 디버깅할 수 있습니다. 예를 들어, Curator가 실패 상태이지만 로그 메시지가 이유를 제공하지 않으면 예약된 다른 cron 작업을 기다리지 않고 로그 레벨을 늘리고 새 작업을 트리거할 수 있습니다.

#### 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

## 프로시저

Curator 디버그 로그를 활성화하고 다음 Curator 반복을 수동으로 트리거하려면 다음을 수행합니다.

- Curator의 디버그 로그를 활성화합니다.

```
$ oc set env cronjob/curator CURATOR_LOG_LEVEL=DEBUG
CURATOR_SCRIPT_LOG_LEVEL=DEBUG
```

로그 수준을 지정합니다.

- CRITICAL.** Curator가 심각한 메시지만 표시합니다.
- ERROR.** Curator가 오류 및 심각한 메시지만 표시합니다.
- WARNING.** Curator가 오류, 경고 및 심각한 메시지만 표시합니다.
- INFO.** Curator가 정보, 오류, 경고 및 심각한 메시지만 표시합니다.
- DEBUG.** Curator가 위의 모든 항목 외에도 디버그 메시지만 표시합니다.  
기본값은 INFO입니다.



### 참고

클러스터 로깅은 OpenShift Container Platform 랩퍼 스크립트(**run.sh** 및 **convert.py**)에서 OpenShift Container Platform 사용자 정의 환경 변수 **CURATOR\_SCRIPT\_LOG\_LEVEL**을 사용합니다. 환경 변수는 필요에 따라 스크립트 디버깅을 위해 **CURATOR\_LOG\_LEVEL**과 동일한 값을 갖습니다.

- 다음 Curator 반복을 트리거합니다.

```
$ oc create job --from=cronjob/curator <job_name>
```

- 다음 명령을 사용하여 CronJob을 제어합니다.

- CronJob 일시 중단:

```
$ oc patch cronjob curator -p '{"spec":{"suspend":true}}'
```

- CronJob 다시 시작:

```
$ oc patch cronjob curator -p '{"spec":{"suspend":false}}'
```

- CronJob 일정 변경:

```
$ oc patch cronjob curator -p '{"spec":{"schedule":"0 0 * * *"}}' 1
```

- 1** **schedule** 옵션은 **cron 형식**의 일정을 승인합니다.

## 10장. 클러스터 로깅 삭제

OpenShift Container Platform 클러스터에서 클러스터 로깅을 제거할 수 있습니다.

### 10.1. OPENSIFT CONTAINER PLATFORM에서 클러스터 로깅 설치 삭제

클러스터에서 클러스터 로깅을 제거할 수 있습니다.

#### 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

#### 프로시저

클러스터 로깅을 제거하려면 다음을 수행합니다.

- 다음 명령을 사용하여 배포 중에 생성된 모든 항목을 제거합니다.

```
$ oc delete clusterlogging instance -n openshift-logging
```

- 다음 명령을 사용하여 Operator 인스턴스가 삭제된 후에도 남아있는 영구 볼륨 클레임을 제거합니다.

```
$ oc delete pvc --all -n openshift-logging
```

## 11장. 내보낸 필드

로깅 시스템에서 내보낸 필드이며 Elasticsearch 및 Kibana에서 검색할 수 있습니다. 검색할 때 점으로 구분된 전체 필드 이름을 사용합니다. 예를 들어 Elasticsearch `/_search` URL의 경우 Kubernetes 포드 이름을 찾으려면 `/_search/q=kubernetes.pod_name:name-of-my-pod`를 사용합니다.

다음 섹션에서는 로깅 저장소에 없을 수 있는 필드에 대해 설명합니다. 모든 레코드에 이러한 필드가 모두 있는 것은 아닙니다. 필드는 다음 범주로 그룹화됩니다.

- **exported-fields-Default**
- **exported-fields-systemd**
- **exported-fields-kubernetes**
- **exported-fields-pipeline\_metadata**
- **exported-fields-ovirt**
- **exported-fields-aushape**
- **exported-fields-tlog**

### 11.1. 기본 내보낸 필드

로깅 시스템에서 내보낸 기본 필드이며 Elasticsearch 및 Kibana에서 검색할 수 있습니다. 기본 필드는 최상위 수준 및 **collectd\***입니다.

#### 최상위 수준 필드

최상위 수준 필드는 모든 애플리케이션에 공통이며 모든 레코드에 존재할 수 있습니다. Elasticsearch 템플릿의 경우 최상위 수준 필드는 템플릿의 매핑 섹션에서 **기본값**의 실제 매핑을 채웁니다.

매개변수	설명
<b>@timestamp</b>	로그 페이로드가 작성되거나 작성 시간을 알 수 없는 경우 로그 페이로드가 처음 수집될 때 표시되는 UTC 값입니다. 이는 로그 페이로드가 생성된 시기에 대한 로그 처리 파이프라인의 최선의 노력으로 이루어진 결정입니다. @ 접두사 규칙을 추가하여 특정 용도로 예약된 필드를 기록합니다. Elasticsearch를 사용하면 대부분의 도구가 기본적으로 <b>@timestamp</b> 를 찾습니다. 예를 들어 형식은 2015-01-24 14:06:05.071000입니다.
<b>geoip</b>	이는 시스템의 geo-ip입니다.
<b>호스트 이름</b>	<b>호스트 이름</b> 은 원래 페이로드를 생성하는 엔티티의 FQDN(정규화된 도메인 이름)입니다. 이 필드는 이 컨텍스트를 도출하려는 시도입니다. 이를 생성하는 엔티티는 컨텍스트를 알고 있는 경우가 있습니다. 다른 경우에는 해당 엔티티에 제한된 네임스페이스 자체가 있으며 이는 수집기 또는 노멀라이저로 알려져 있습니다.
<b>ipaddr4</b>	소스 서버의 IP 주소 V4. 배열될 수 있습니다.
<b>ipaddr6</b>	사용 가능한 경우 소스 서버의 IP 주소 V6입니다.

매개변수	설명
<b>level</b>	<p>Python의 로깅 모듈인 <code>rsyslog(severitytext 속성)</code>에서 제공하는 로깅 수준입니다. 가능한 값은 <a href="#">misc/sys/syslog.h</a>와 <code>trace</code> 및 <code>unknown</code>에 나와 있습니다. 예를 들면 "alert crit debug emerg err info notice trace unknown warning"입니다. <code>trace</code>는 <code>syslog.h</code> 목록에 없지만 많은 애플리케이션에서 사용합니다.</p> <p>.로깅 시스템이 이해하지 못하는 값을 얻을 때만 <code>unknown</code> 값을 사용해야 하며 최고 수준임을 유의하십시오.. <code>debug</code>보다 자세한 높은 이상으로 <code>trace</code>를 고려하십시오.. <code>error</code>는 더 이상 사용되지 않으며 <code>err</code>를 사용합니다. <code>panic</code>을 <code>emerg</code>로 변환합니다. <code>warn</code>을 <code>warning</code>으로 변환합니다.</p> <p><code>syslog/journal PRIORITY</code>의 숫자 값은 일반적으로 <a href="#">misc/sys/syslog.h</a>에 나열된 우선 순위 값을 사용하여 매핑할 수 있습니다.</p> <p>다른 로깅 시스템의 로그 수준과 우선 순위는 가장 가까운 일치 항목에 매핑해야 합니다. 예제는 <a href="#">python 로깅</a>을 참조하십시오.</p>
<b>message</b>	일반적인 로그 항목 메시지 또는 페이로드입니다. 수집기 또는 노멀라이저, 즉 UTF-8로 인코딩된 메타 데이터를 제거할 수 있습니다.
<b>pid</b>	사용 가능한 경우 이는 로깅 엔티티의 프로세스 ID입니다.
<b>service</b>	사용 가능한 경우 로깅 엔티티와 연관된 서비스의 이름입니다. 예를 들어 <code>syslog APP-NAME</code> 특성은 서비스 필드에 매핑됩니다.
<b>tags</b>	선택적으로 수집기 또는 노멀라이저가 각 로그에 배치한 Operator 정의 태그 목록을 제공합니다. 페이로드는 공백으로 구분된 문자열 토큰이 있는 문자열이거나 문자열 토큰의 JSON 목록일 수 있습니다.
<b>file</b>	파일 경로에 대한 수집기 <b>TODO</b> 분석기에 로컬인 로그 항목을 포함하는 파일의 선택적 경로입니다.
<b>offset</b>	오프셋 값은 단일 로그 파일의 컨텍스트에서 값이 엄격하게 단조롭게 증가하는 한 파일에서 로그 라인의 시작까지의 바이트 수(0 또는 1기반) 또는 로그 라인 번호(0 또는 1기반)를 나타낼 수 있습니다. 새 버전의 로그 파일(회전)을 나타내는 값을 줄바꿈할 수 있습니다.
<b>namespace_name</b>	이 레코드를 이름을 공유하는 <b>네임스페이스</b> 와 연결합니다. 이 값은 저장되지 않지만 레코드를 액세스 제어 및 시각화를 위한 적절한 <b>네임스페이스</b> 와 연결하는 데 사용됩니다. 일반적으로 이 값은 태그에 제공되지만 프로토콜이 태그 전송을 지원하지 않으면 이 필드를 사용할 수 있습니다. 이 필드가 있으면 태그 또는 <code>kubernetes.namespace_name</code> 에 지정된 <b>네임스페이스</b> 를 대체합니다.
<b>namespace_uuid</b>	이는 <code>namespace_name</code> 과 연관된 <b>uuid</b> 입니다. 이 값은 저장되지 않지만 레코드를 액세스 제어 및 시각화를 위한 적절한 <b>네임스페이스</b> 와 연결하는 데 사용됩니다. 이 필드가 있으면 <code>kubernetes.namespace_uuid</code> 에 지정된 <b>uuid</b> 를 대체합니다. 또한 이 로그 레코드에 대해 Kubernetes 메타 데이터 조회를 건너 뛩니다.

**collectd 필드**

다음 필드는 네임스페이스 지표 메타 데이터를 나타냅니다.

매개 변수	설명
<b>collectd.interval</b>	유형: 플로트 <b>collectd</b> 간격.
<b>collectd.plugin</b>	유형: 문자열 <b>collectd</b> 플러그인.
<b>collectd.plugin_instance</b>	유형: 문자열 <b>collectd</b> plugin_instance.
<b>collectd.type_instance</b>	유형: 문자열 <b>collectd</b> type_instance.
<b>collectd.type</b>	유형: 문자열 <b>collectd</b> 유형.
<b>collectd.dtypes</b>	유형: 문자열 <b>collectd</b> dtypes.

**collectd.processes 필드**

다음 필드는 **collectd** 프로세스 플러그인에 해당합니다.

매개 변수	설명
<b>collectd.processes.ps_state</b>	유형: 정수 프로세스 플러그인의 <b>collectd ps_state</b> 유형.

**collectd.processes.ps\_disk\_ops 필드**

프로세스 플러그인의 **collectd ps\_disk\_ops** 유형.

매개 변수	설명
<b>collectd.processes.ps_disk_ops.read</b>	유형: 플로트 <b>TODO</b>
<b>collectd.processes.ps_disk_ops.write</b>	유형: 플로트 <b>TODO</b>



매개 변수	설명
<b>collectd.processes.ps_vm</b>	유형: 정수 프로세스 플러그인의 <b>collectd ps_vm</b> 유형.
<b>collectd.processes.ps_rs</b>	유형: 정수 프로세스 플러그인의 <b>collectd ps_rss</b> 유형.
<b>collectd.processes.ps_data</b>	유형: 정수 프로세스 플러그인의 <b>collectd ps_data</b> 유형.
<b>collectd.processes.ps_code</b>	유형: 정수 프로세스 플러그인의 <b>collectd ps_code</b> 유형.
<b>collectd.processes.ps_stacksize</b>	유형: 정수 프로세스 플러그인의 <b>collectd ps_stacksize</b> 유형.

**collectd.processes.ps\_cputime** 필드  
프로세스 플러그인의 **collectd ps\_cputime** 유형.

매개 변수	설명
<b>collectd.processes.ps_cputime.user</b>	유형: 플로트 <b>TODO</b>
<b>collectd.processes.ps_cputime.syst</b>	유형: 플로트 <b>TODO</b>

**collectd.processes.ps\_count** 필드  
프로세스 플러그인의 **collectd ps\_count** 유형.

매개 변수	설명
<b>collectd.processes.ps_count.processes</b>	유형: 정수 <b>TODO</b>
<b>collectd.processes.ps_count.threads</b>	유형: 정수 <b>TODO</b>

**collectd.processes.ps\_pagefaults** 필드

프로세스 플러그인의 **collectd ps\_pagefaults** 유형.

매개 변수	설명
<b>collectd.processes.ps_pagefaults.majflt</b>	유형: 플롯 <b>TODO</b>
<b>collectd.processes.ps_pagefaults.minflt</b>	유형: 플롯 <b>TODO</b>

**collectd.processes.ps\_disk\_octets** 필드

프로세스 플러그인의 **collectd ps\_disk\_octets** 유형.

매개 변수	설명
<b>collectd.processes.ps_disk_octets.read</b>	유형: 플롯 <b>TODO</b>
<b>collectd.processes.ps_disk_octets.write</b>	유형: 플롯 <b>TODO</b>
<b>collectd.processes.fork_rate</b>	유형: 플롯 프로세스 플러그인의 <b>collectd fork_rate</b> 유형.

**collectd.disk** 필드

**collectd** 디스크 플러그인에 해당합니다.

**collectd.disk.disk\_merged** 필드

디스크 플러그인의 **collectd disk\_merged** 유형.

매개 변수	설명
<b>collectd.disk.disk_merged.read</b>	유형: 플롯 <b>TODO</b>
<b>collectd.disk.disk_merged.write</b>	유형: 플롯 <b>TODO</b>

**collectd.disk.disk\_octets** 필드

디스크 플러그인의 **collectd disk\_octets** 유형.

매개 변수	설명
<b>collectd.disk.disk_octets.read</b>	유형: 플롯 <b>TODO</b>
<b>collectd.disk.disk_octets.write</b>	유형: 플롯 <b>TODO</b>

**collectd.disk.disk\_time** 필드  
디스크 플러그인의 **collectd disk\_time** 유형.

매개 변수	설명
<b>collectd.disk.disk_time.read</b>	유형: 플롯 <b>TODO</b>
<b>collectd.disk.disk_time.write</b>	유형: 플롯 <b>TODO</b>

**collectd.disk.disk\_ops** 필드  
디스크 플러그인의 **collectd disk\_ops** 유형.

매개 변수	설명
<b>collectd.disk.disk_ops.read</b>	유형: 플롯 <b>TODO</b>
<b>collectd.disk.disk_ops.write</b>	유형: 플롯 <b>TODO</b>
<b>collectd.disk.pending_operations</b>	유형: 정수 디스크 플러그인의 <b>collectd pending_operations</b> 유형.

**collectd.disk.disk\_io\_time** 필드  
디스크 플러그인의 **collectd disk\_io\_time** 유형.

매개 변수	설명
<b>collectd.disk.disk_io_time.io_time</b>	유형: 플롯 <b>TODO</b>

매개 변수	설명
<b>collectd.disk.disk_io_time_weighted_io_time</b>	유형: 플롯 <b>TODO</b>

**collectd.interface 필드**

**collectd** 인터페이스 플러그인에 해당합니다.

**collectd.interface.if\_octets 필드**

인터페이스 플러그인의 **collectd if\_octets** 유형.

매개 변수	설명
<b>collectd.interface.if_octets.rx</b>	유형: 플롯 <b>TODO</b>
<b>collectd.interface.if_octets.tx</b>	유형: 플롯 <b>TODO</b>

**collectd.interface.if\_packets 필드**

인터페이스 플러그인의 **collectd if\_packets** 유형.

매개 변수	설명
<b>collectd.interface.if_packets.rx</b>	유형: 플롯 <b>TODO</b>
<b>collectd.interface.if_packets.tx</b>	유형: 플롯 <b>TODO</b>

**collectd.interface.if\_errors 필드**

인터페이스 플러그인의 **collectd if\_errors** 유형.

매개 변수	설명
<b>collectd.interface.if_errors.rx</b>	유형: 플롯 <b>TODO</b>
<b>collectd.interface.if_errors.tx</b>	유형: 플롯 <b>TODO</b>

**collectd.interface.if\_dropped 필드**

인터페이스 플러그인의 **collectd if\_dropped** 유형.

매개 변수	설명
<b>collectd.interface.if_dropped.rx</b>	유형: 플로트 <b>TODO</b>
<b>collectd.interface.if_dropped.tx</b>	유형: 플로트 <b>TODO</b>

**collectd.virt 필드**

**collectd** 가상화 플러그인에 해당합니다.

**collectd.virt.if\_octets 필드**

가상화 플러그인의 **collectd if\_octets** 유형.

매개 변수	설명
<b>collectd.virt.if_octets.rx</b>	유형: 플로트 <b>TODO</b>
<b>collectd.virt.if_octets.tx</b>	유형: 플로트 <b>TODO</b>

**collectd.virt.if\_packets 필드**

가상화 플러그인의 **collectd if\_packets** 유형.

매개 변수	설명
<b>collectd.virt.if_packets.rx</b>	유형: 플로트 <b>TODO</b>
<b>collectd.virt.if_packets.tx</b>	유형: 플로트 <b>TODO</b>

**collectd.virt.if\_errors 필드**

가상화 플러그인의 **collectd if\_errors** 유형.

매개 변수	설명
-------	----

매개 변수	설명
<b>collectd.virt.if_errors.rx</b>	유형: 플롯 <b>TODO</b>
<b>collectd.virt.if_errors.tx</b>	유형: 플롯 <b>TODO</b>

**collectd.virt.if\_dropped** 필드  
가상화 플러그인의 **collectd if\_dropped** 유형.

매개 변수	설명
<b>collectd.virt.if_dropped.rx</b>	유형: 플롯 <b>TODO</b>
<b>collectd.virt.if_dropped.tx</b>	유형: 플롯 <b>TODO</b>

**collectd.virt.disk\_ops** 필드  
가상화 플러그인의 **collectd disk\_ops** 유형.

매개 변수	설명
<b>collectd.virt.disk_ops.read</b>	유형: 플롯 <b>TODO</b>
<b>collectd.virt.disk_ops.write</b>	유형: 플롯 <b>TODO</b>

**collectd.virt.disk\_octets** 필드  
가상화 플러그인의 **collectd disk\_octets** 유형.

매개 변수	설명
<b>collectd.virt.disk_octets.read</b>	유형: 플롯 <b>TODO</b>
<b>collectd.virt.disk_octets.write</b>	유형: 플롯 <b>TODO</b>

매개변수	설명
<b>collectd.virt.memory</b>	유형: 플로트 가상화 플러그인의 <b>collectd</b> 메모리 유형.
<b>collectd.virt.virt_vcpu</b>	유형: 플로트 가상화 플러그인의 <b>collectd virt_vcpu</b> 유형.
<b>collectd.virt.virt_cpu_total</b>	유형: 플로트 가상화 플러그인의 <b>collectd virt_cpu_total</b> 유형.

**collectd.CPU 필드**

**collectd** CPU 플러그인에 해당합니다.

매개변수	설명
<b>collectd.CPU.percent</b>	유형: 플로트 플러그인 CPU의 <b>collectd</b> 유형 백분율.

**collectd.df 필드**

**collectd df** 플러그인에 해당합니다.

매개변수	설명
<b>collectd.df.df_complex</b>	유형: 플로트 플러그인 <b>df</b> 의 <b>collectd</b> 유형 <b>df_complex</b> .
<b>collectd.df.percent_bytes</b>	유형: 플로트 플러그인 <b>df</b> 의 <b>collectd</b> 유형 <b>percent_bytes</b> .

**collectd.entropy 필드**

**collectd** 엔트로피 플러그인에 해당합니다.

매개변수	설명
<b>collectd.entropy.entropy</b>	유형: 정수 엔트로피 플러그인의 <b>collectd</b> 엔트로피 유형.

**collectd.memory 필드**

**collectd** 메모리 플러그인에 해당합니다.

매개 변수	설명
<b>collectd.memory.memory</b>	<p>유형: 플로트</p> <p>메모리 플러그인의 <b>collectd</b> 메모리 유형.</p>
<b>collectd.memory.percent</b>	<p>유형: 플로트</p> <p>메모리 플러그인의 <b>collectd</b> 백분율 유형.</p>

**collectd.swap** 필드

**collectd** 스왑 플러그인에 해당합니다.

매개 변수	설명
<b>collectd.swap.swap</b>	<p>유형: 정수</p> <p>스왑 플러그인의 <b>collectd</b> 스왑 유형.</p>
<b>collectd.swap.swap_io</b>	<p>유형: 정수</p> <p>스왑 플러그인의 <b>collectd swap_io</b> 유형.</p>

**collectd.load** 필드

**collectd** 로드 플러그인에 해당합니다.

**collectd.load.load** 필드

로드 플러그인의 **collectd** 로드 유형

매개 변수	설명
<b>collectd.load.load.shortterm</b>	<p>유형: 플로트</p> <p><b>TODO</b></p>
<b>collectd.load.load.midterm</b>	<p>유형: 플로트</p> <p><b>TODO</b></p>
<b>collectd.load.load.longterm</b>	<p>유형: 플로트</p> <p><b>TODO</b></p>

**collectd.aggregation** 필드

**collectd** 집계 플러그인에 해당합니다.



매개 변수	설명
<b>collectd.aggregation.percent</b>	유형: 플로트 <b>TODO</b>

**collectd.statsd 필드**

**collectd statsd** 플러그인에 해당합니다.

매개 변수	설명
<b>collectd.statsd.host_cpu</b>	유형: 정수 <b>statsd</b> 플러그인의 <b>collectd</b> CPU 유형.
<b>collectd.statsd.host_elapsed_time</b>	유형: 정수 <b>statsd</b> 플러그인의 <b>collectd elapsed_time</b> 유형.
<b>collectd.statsd.host_memory</b>	유형: 정수 <b>statsd</b> 플러그인의 <b>collectd</b> 메모리 유형.
<b>collectd.statsd.host_nic_speed</b>	유형: 정수 <b>statsd</b> 플러그인의 <b>collectd nic_speed</b> 유형.
<b>collectd.statsd.host_nic_rx</b>	유형: 정수 <b>statsd</b> 플러그인의 <b>collectd nic_rx</b> 유형.
<b>collectd.statsd.host_nic_tx</b>	유형: 정수 <b>statsd</b> 플러그인의 <b>collectd nic_tx</b> 유형.
<b>collectd.statsd.host_nic_rx_dropped</b>	유형: 정수 <b>statsd</b> 플러그인의 <b>collectd nic_rx_dropped</b> 유형.
<b>collectd.statsd.host_nic_tx_dropped</b>	유형: 정수 <b>statsd</b> 플러그인의 <b>collectd nic_tx_dropped</b> 유형.
<b>collectd.statsd.host_nic_rx_errors</b>	유형: 정수 <b>statsd</b> 플러그인의 <b>collectd nic_rx_errors</b> 유형.
<b>collectd.statsd.host_nic_tx_errors</b>	유형: 정수 <b>statsd</b> 플러그인의 <b>collectd nic_tx_errors</b> 유형.

매개 변수	설명
<b>collectd.statsd.host_storage</b>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd</b> 스토리지 유형.</p>
<b>collectd.statsd.host_swap</b>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd</b> 스왑 유형.</p>
<b>collectd.statsd.host_vdsm</b>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd</b> VDSM 유형.</p>
<b>collectd.statsd.host_vms</b>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd</b> VMS 유형.</p>
<b>collectd.statsd.vm_nic_tx_dropped</b>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd nic_tx_dropped</b> 유형.</p>
<b>collectd.statsd.vm_nic_rx_bytes</b>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd nic_rx_bytes</b> 유형.</p>
<b>collectd.statsd.vm_nic_tx_bytes</b>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd nic_tx_bytes</b> 유형.</p>
<b>collectd.statsd.vm_balloon_min</b>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd balloon_min</b> 유형.</p>
<b>collectd.statsd.vm_balloon_max</b>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd balloon_max</b> 유형.</p>
<b>collectd.statsd.vm_balloon_target</b>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd balloon_target</b> 유형.</p>
<b>collectd.statsd.vm_balloon_cur</b>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd balloon_cur</b> 유형.</p>
<b>collectd.statsd.vm_cpu_sys</b>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd cpu_sys</b> 유형.</p>

매개 변수	설명
<code>collectd.statsd.vm_cpu_usage</code>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd cpu_usage</b> 유형.</p>
<code>collectd.statsd.vm_disk_read_ops</code>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd disk_read_ops</b> 유형.</p>
<code>collectd.statsd.vm_disk_write_ops</code>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd disk_write_ops</b> 유형.</p>
<code>collectd.statsd.vm_disk_flush_latency</code>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd disk_flush_latency</b> 유형.</p>
<code>collectd.statsd.vm_disk_apparent_size</code>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd disk_apparent_size</b> 유형.</p>
<code>collectd.statsd.vm_disk_write_bytes</code>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd disk_write_bytes</b> 유형.</p>
<code>collectd.statsd.vm_disk_write_rate</code>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd disk_write_rate</b> 유형.</p>
<code>collectd.statsd.vm_disk_true_size</code>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd disk_true_size</b> 유형.</p>
<code>collectd.statsd.vm_disk_read_rate</code>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd disk_read_rate</b> 유형.</p>
<code>collectd.statsd.vm_disk_write_latency</code>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd disk_write_latency</b> 유형.</p>
<code>collectd.statsd.vm_disk_read_latency</code>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd disk_read_latency</b> 유형.</p>
<code>collectd.statsd.vm_disk_read_bytes</code>	<p>유형: 정수</p> <p><b>statsd</b> 플러그인의 <b>collectd disk_read_bytes</b> 유형.</p>

매개 변수	설명
<code>collectd.statsd.vm_nic_rx_dropped</code>	유형: 정수 <b>statsd</b> 플러그인의 <b>collectd nic_rx_dropped</b> 유형.
<code>collectd.statsd.vm_cpu_user</code>	유형: 정수 <b>statsd</b> 플러그인의 <b>collectd cpu_user</b> 유형.
<code>collectd.statsd.vm_nic_rx_errors</code>	유형: 정수 <b>statsd</b> 플러그인의 <b>collectd nic_rx_errors</b> 유형.
<code>collectd.statsd.vm_nic_tx_errors</code>	유형: 정수 <b>statsd</b> 플러그인의 <b>collectd nic_tx_errors</b> 유형.
<code>collectd.statsd.vm_nic_speed</code>	유형: 정수 <b>statsd</b> 플러그인의 <b>collectd nic_speed</b> 유형.

**collectd.postgresql** 필드  
**collectd postgresql** 플러그인에 해당합니다.

매개 변수	설명
<code>collectd.postgresql.pg_n_tup_g</code>	유형: 정수 플러그인 postgresql의 <b>collectd</b> 유형 <b>pg_n_tup_g</b> .
<code>collectd.postgresql.pg_n_tup_c</code>	유형: 정수 플러그인 postgresql의 <b>collectd</b> 유형 <b>pg_n_tup_c</b> .
<code>collectd.postgresql.pg_n_umbakends</code>	유형: 정수 플러그인 postgresql의 <b>collectd</b> 유형 <b>pg_numbackends</b> .
<code>collectd.postgresql.pg_xact</code>	유형: 정수 플러그인 postgresql의 <b>collectd</b> 유형 <b>pg_xact</b> .
<code>collectd.postgresql.pg_db_size</code>	유형: 정수 플러그인 postgresql의 <b>collectd</b> 유형 <b>pg_db_size</b> .
<code>collectd.postgresql.pg_blks</code>	유형: 정수 플러그인 postgresql의 <b>collectd</b> 유형 <b>pg_blks</b> .

## 11.2. SYSTEMD 내보낸 필드

이는 Elasticsearch와 Kibana에서 검색에 사용할 수 있는 OpenShift Container Platform 클러스터 로깅에서 보낸 **systemd** 필드입니다.

**systemd** 저널에 공통 필드의 특징을 포함합니다. **애플리케이션**은 자체 필드를 저널에 쓸 수 있습니다. 이는 **systemd.u** 네임스페이스에서 사용할 수 있습니다. **RESULT** 및 **UNIT**은 이러한 두 가지 필드입니다.

### systemd.k 필드

다음 표에는 **systemd** 커널 특정 메타 데이터가 포함되어 있습니다.

매개 변수	설명
<b>systemd.k.KERNEL_DEVICE</b>	<b>systemd.k.KERNEL_DEVICE</b> 는 커널 장치 이름입니다.
<b>systemd.k.KERNEL_SUBSYSTEM</b>	<b>systemd.k.KERNEL_SUBSYSTEM</b> 은 커널 하위 시스템 이름입니다.
<b>systemd.k.UDEV_DEVLINK</b>	<b>systemd.k.UDEV_DEVLINK</b> 에는 노드를 가리키는 추가 기호 링크 이름이 포함되어 있습니다.
<b>systemd.k.UDEV_DEVNODE</b>	<b>systemd.k.UDEV_DEVNODE</b> 는 장치의 노드 경로입니다.
<b>systemd.k.UDEV_SYSNAME</b>	<b>systemd.k.UDEV_SYSNAME</b> 은 커널 장치 이름입니다.

### systemd.t 필드

**systemd.t** 필드는 신뢰할 수 있는 저널 필드, 저널이 암시적으로 추가한 필드이며 클라이언트 코드로 변경할 수 없습니다.

매개 변수	설명
<b>systemd.t.AUDIT_LOGINUID</b>	<b>systemd.t.AUDIT_LOGINUID</b> 는 저널 프로세스의 사용자 ID입니다.
<b>systemd.t.BOOT_ID</b>	<b>systemd.t.BOOT_ID</b> 는 커널 부팅 ID입니다.
<b>systemd.t.AUDIT_SESSION</b>	<b>systemd.t.AUDIT_SESSION</b> 은 저널 프로세스의 세션입니다.
<b>systemd.t.CAP_EFFECTIVE</b>	<b>systemd.t.CAP_EFFECTIVE</b> 는 저널 프로세스의 기능을 나타냅니다.
<b>systemd.t.CMDLINE</b>	<b>systemd.t.CMDLINE</b> 은 저널 프로세스의 명령줄입니다.
<b>systemd.t.COMM</b>	<b>systemd.t.COMM</b> 은 저널 항목 프로세스의 이름입니다.

매개변수	설명
<b>systemd.t.EXE</b>	<b>systemd.t.EXE</b> 는 저널 항목 프로세스의 실행 가능 경로입니다.
<b>systemd.t.GID</b>	<b>systemd.t.GID</b> 는 저널 항목 프로세스의 그룹 ID입니다.
<b>systemd.t.HOSTNAME</b>	<b>systemd.t.HOSTNAME</b> 은 호스트 이름입니다.
<b>systemd.t.MACHINE_ID</b>	<b>systemd.t.MACHINE_ID</b> 는 호스트의 컴퓨터 ID입니다.
<b>systemd.t.PID</b>	<b>systemd.t.PID</b> 는 저널 항목 프로세스의 프로세스 ID입니다.
<b>systemd.t.SELINUX_CONTEXT</b>	<b>systemd.t.SELINUX_CONTEXT</b> 는 저널 프로세스의 보안 컨텍스트 또는 레이블입니다.
<b>systemd.t.SOURCE_REALTIME_TIMESTAMP</b>	<b>systemd.t.SOURCE_REALTIME_TIMESTAMP</b> 는 메시지의 가장 빠르고 가장 안정적인 타임 스탬프입니다. 이는 RFC 3339 NS 형식으로 변환됩니다.
<b>systemd.t.SYSTEMD_CGROUP</b>	<b>systemd.t.SYSTEMD_CGROUP</b> 는 <b>systemd</b> 제어 그룹 경로입니다.
<b>systemd.t.SYSTEMD_OWNER_UID</b>	<b>systemd.t.SYSTEMD_OWNER_UID</b> 는 세션의 소유자 ID입니다.
<b>systemd.t.SYSTEMD_SESSION</b>	해당되는 경우 <b>systemd.t.SYSTEMD_SESSION</b> 은 <b>systemd</b> 세션 ID입니다.
<b>systemd.t.SYSTEMD_SLICE</b>	<b>systemd.t.SYSTEMD_SLICE</b> 는 저널 프로세스의 슬라이스 단위입니다.
<b>systemd.t.SYSTEMD_UNIT</b>	<b>systemd.t.SYSTEMD_UNIT</b> 는 세션의 단위 이름입니다.
<b>systemd.t.SYSTEMD_USER_UNIT</b>	해당되는 경우 <b>systemd.t.SYSTEMD_USER_UNIT</b> 는 세션의 사용자 단위 이름입니다.
<b>systemd.t.TRANSPORT</b>	<b>systemd.t.TRANSPORT</b> 는 저널 서비스의 입력 방법입니다. 여기에는 <b>audit</b> , <b>driver</b> , <b>syslog</b> , <b>journal</b> , <b>stdout</b> 및 <b>kernel</b> 이 포함됩니다.
<b>systemd.t.UID</b>	<b>systemd.t.UID</b> 는 저널 프로세스의 사용자 ID입니다.
<b>systemd.t.SYSLOG_FACILITY</b>	<b>systemd.t.SYSLOG_FACILITY</b> 는 <b>syslog</b> 에 대한 기능을 포함하는 필드이며 10진수 문자열 형식입니다.
<b>systemd.t.SYSLOG_IDENTIFIER</b>	<b>systemd.t.systemd.t.SYSLOG_IDENTIFIER</b> 는 <b>syslog</b> 의 식별자입니다.

매개변수	설명
<b>systemd.t.SYSLOG_PID</b>	<b>SYSLOG_PID</b> 는 <b>syslog</b> 의 클라이언트 프로세스 ID입니다.

**systemd.u 필드**

**systemd.u** 필드는 클라이언트에서 직접 전달되어 저널에 저장됩니다.

매개변수	설명
<b>systemd.u.CODE_FILE</b>	<b>systemd.u.CODE_FILE</b> 은 소스의 파일 이름이 포함된 코드 위치입니다.
<b>systemd.u.CODE_FUNCTION</b>	<b>systemd.u.CODE_FUNCTION</b> 은 소스 기능을 포함하는 코드 위치입니다.
<b>systemd.u.CODE_LINE</b>	<b>systemd.u.CODE_LINE</b> 은 소스의 라인 번호를 포함하는 코드 위치입니다.
<b>systemd.u.ERRNO</b>	<b>systemd.u.ERRNO</b> 는(있는 경우) 10진수 문자열로 숫자 값 형식의 하위 수준 오류 번호입니다.
<b>systemd.u.MESSAGE_ID</b>	<b>systemd.u.MESSAGE_ID</b> 는 메시지 유형을 인식하기 위한 메시지 식별자 ID입니다.
<b>systemd.u.RESULT</b>	개인 용도로만 사용하십시오.
<b>systemd.u.UNIT</b>	개인 용도로만 사용하십시오.

**11.3. KUBERNETES 내보낸 필드**

다음은 Elasticsearch 및 Kibana에서 검색할 수 있는 OpenShift Container Platform 클러스터 로깅으로 내보낸 Kubernetes 필드입니다.

Kubernetes 관련 메타데이터의 네임스페이스입니다. **kubernetes.pod\_name**은 포드의 이름입니다.

**kubernetes.labels 필드**

OpenShift 오브젝트에 부착된 레이블은 **kubernetes.labels**입니다. 각 레이블 이름은 레이블 필드의 하위 필드입니다. 각 레이블 이름은 점으로 구분되어 있으므로 이름의 점이 밑줄로 바뀝니다.

매개변수	설명
<b>kubernetes.pod_id</b>	포드의 Kubernetes ID입니다.
<b>kubernetes.namespace_name</b>	Kubernetes의 네임스페이스 이름입니다.
<b>kubernetes.namespace_id</b>	Kubernetes의 네임스페이스 ID입니다.

매개변수	설명
<b>kubernetes.host</b>	Kubernetes 노드 이름입니다.
<b>kubernetes.container_name</b>	Kubernetes의 컨테이너 이름입니다.
<b>kubernetes.labels.deployment</b>	Kubernetes 오브젝트와 관련된 배포입니다.
<b>kubernetes.labels.deploymentconfig</b>	Kubernetes 오브젝트와 관련된 배포 구성입니다.
<b>kubernetes.labels.component</b>	Kubernetes 오브젝트와 관련된 구성 요소입니다.
<b>kubernetes.labels.provider</b>	Kubernetes 오브젝트와 관련된 공급자입니다.

**kubernetes.annotations 필드**

OpenShift 오브젝트와 관련된 주석은 **kubernetes.annotations** 필드입니다.

**11.4. 컨테이너 내보낸 필드**

Elasticsearch 및 Kibana에서 검색할 수 있는 OpenShift Container Platform 클러스터 로깅으로 내보낸 Docker 필드입니다. Docker 컨테이너 특정 메타데이터의 네임스페이스입니다. `docker.container_id`는 Docker 컨테이너 ID입니다.

**pipeline\_metadata.collector 필드**

이 섹션에는 수집기 고유의 메타데이터가 포함되어 있습니다.

매개변수	설명
<b>pipeline_metadata.collector.hostname</b>	수집기의 FQDN입니다. 실제 로그 송신기의 FQDN과 다를 수 있습니다.
<b>pipeline_metadata.collector.name</b>	수집기의 이름입니다.
<b>pipeline_metadata.collector.version</b>	수집기의 버전입니다.
<b>pipeline_metadata.collector.ipaddr4</b>	수집기 서버의 IP 주소 v4는 배열일 수 있습니다.
<b>pipeline_metadata.collector.ipaddr6</b>	수집기 서버의 IP 주소 v6은 배열일 수 있습니다.



매개 변수	설명
<b>pipeline_metadata.collect or.inputname</b>	TCP/UDP이든 imjournal/imfile이든 수집기에서 로그 메시지를 수신한 방법입니다.
<b>pipeline_metadata.collect or.received_at</b>	수집기가 메시지를 수신한 시간입니다.
<b>pipeline_metadata.collect or.original_raw_message</b>	수집기에서 수집하거나 가능한 한 소스에 가까운 구문 분석되지 않은 원본 로그 메시지입니다.

### pipeline\_metadata.normalizer 필드

이 섹션에는 노멀라이저 고유의 메타 이터가 포함되어 있습니다.

매개 변수	설명
<b>pipeline_metadata.normal izer.hostname</b>	노멀라이저의 FQDN입니다.
<b>pipeline_metadata.normal izer.name</b>	노멀라이저의 이름입니다.
<b>pipeline_metadata.normal izer.version</b>	노멀라이저의 버전입니다.
<b>pipeline_metadata.normal izer.ipaddr4</b>	노멀라이저 서버의 IP 주소 v4는 배열일 수 있습니다.
<b>pipeline_metadata.normal izer.ipaddr6</b>	노멀라이저 서버의 IP 주소 v6은 배열일 수 있습니다.
<b>pipeline_metadata.normal izer.inputname</b>	TCP/UDP인지 여부에 따라 노멀라이저가 로그 메시지를 수신한 방법입니다.
<b>pipeline_metadata.normal izer.received_at</b>	노멀라이저가 메시지를 수신한 시간입니다.
<b>pipeline_metadata.normal izer.original_raw_messag e</b>	노멀라이저가 수신한 구문 분석되지 않은 원래 로그 메시지입니다.
<b>pipeline_metadata.trace</b>	이 필드는 메시지 추적을 기록합니다. 각 수집기 및 노멀라이저는 자체 정보와 메시지가 처리된 날짜 및 시간을 추가합니다.

## 11.5. OVIRT 내보낸 필드

Elasticsearch 및 Kibana에서 검색할 수 있는 OpenShift Container Platform 클러스터 로깅으로 내보낸 oVirt 필드입니다.

oVirt 메타데이터의 네임스페이스입니다.

매개 변수	설명
<b>ovirt.entity</b>	데이터 소스, 호스트, VMS 및 엔진의 유형입니다.
<b>ovirt.host_id</b>	oVirt 호스트 UUID입니다.

#### ovirt.engine 필드

oVirt 엔진 관련 메타데이터의 네임스페이스입니다. oVirt 엔진의 FQDN은 **ovirt.engine.fqdn**입니다.

## 11.6. AUSHAPE 내보낸 필드

Elasticsearch 및 Kibana에서 검색할 수 있는 OpenShift Container Platform 클러스터 로깅으로 내보낸 Aushape 필드입니다.

Aushape로 변환된 감사 이벤트입니다. 자세한 내용은 [Aushape](#)를 참조하십시오.

매개 변수	설명
<b>aushape.serial</b>	감사 이벤트 일련 번호입니다.
<b>aushape.node</b>	감사 이벤트가 발생한 호스트의 이름입니다.
<b>aushape.error</b>	이벤트를 변환하는 중에 오류가 발생했습니다.
<b>aushape.trimmed</b>	이벤트 크기 제한의 결과로 콘텐츠가 제거된 오브젝트 또는 배열을 지정하여 이벤트 오브젝트에 상대적인 JSONPath 표현식의 배열입니다. 빈 문자열은 이벤트가 내용을 제거했음을 의미하고 빈 배열은 지정되지 않은 오브젝트 및 배열에 의해 트리밍이 발생했음을 의미합니다.
<b>aushape.text</b>	원래 감사 이벤트를 나타내는 배열 로그 레코드 문자열입니다.

#### aushape.data 필드

Aushape와 관련된 감사 이벤트 데이터를 구문 분석했습니다.

매개 변수	설명
<b>aushape.data.avc</b>	유형: 중첩
<b>aushape.data.execve</b>	유형: 문자열
<b>aushape.data.netfilter_cfg</b>	유형: 중첩
<b>aushape.data.obj_pid</b>	유형: 중첩
<b>aushape.data.path</b>	유형: 중첩

## 11.7. TLOG 내보낸 필드

OpenShift Container Platform 클러스터 로깅 시스템에서 내보낸 Tlog 필드는 Elasticsearch 및 Kibana에서 검색할 수 있습니다.

Tlog 터미널 I/O 기록 메시지입니다. 자세한 내용은 [Tlog](#)를 참조하십시오.

매개변수	설명
<b>tlog.ver</b>	메시지 형식 버전 번호입니다.
<b>tlog.user</b>	기록된 사용자 이름입니다.
<b>tlog.term</b>	터미널 유형 이름입니다.
<b>tlog.session</b>	기록된 세션의 감사 세션 ID입니다.
<b>tlog.id</b>	세션 내 메시지의 ID입니다.
<b>tlog.pos</b>	세션에서 메시지 위치입니다(밀리초).
<b>tlog.timing</b>	이 메시지의 이벤트를 제시간에 분배합니다.
<b>tlog.in_txt</b>	유효하지 않은 문자가 스크랩한 텍스트를 입력합니다.
<b>tlog.in_bin</b>	유효하지 않은 입력 문자를 바이트로 스크랩했습니다.
<b>tlog.out_txt</b>	잘못된 문자가 스크랩된 출력 텍스트입니다.
<b>tlog.out_bin</b>	잘못된 출력 문자를 바이트로 스크랩했습니다.