OpenShift Container Platform 4.6

로깅

OpenShift Container Platform에서 클러스터 로깅 구성

Last Updated: 2023-06-23

OpenShift Container Platform 4.6 로깅

OpenShift Container Platform에서 클러스터 로깅 구성

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

법적 공지

초록

이 문서에서는 다양한 OpenShift Container Platform 서비스에 대한 로그를 집계하는 클러스터 로깅의 설치, 구성 및 사용 방법을 설명합니다.

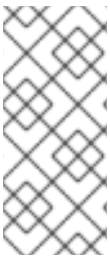
차례

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. Kibana 웹 콘솔을 사용하여 로그 데이터를 시각화할 수 있습니다.

클러스터 기록 작업에서는 다음 유형의 로그를 집계합니다.

- **application** - 인프라 컨테이너 애플리케이션을 제외하고 클러스터에서 실행 중인 사용자 애플리케이션에 의해 생성된 컨테이너 로그입니다.
- **infrastructure** - 저널 로그와 같이 클러스터 및 OpenShift Container Platform 노드에서 실행되는 인프라 구성 요소에서 생성된 로그입니다. 인프라 구성 요소는 **openshift***, **kube*** 또는 **default** 프로젝트에서 실행되는 Pod입니다.
- **audit** - **/var/log/audit/audit.log** 파일에 저장되는 노드 감사 시스템(auditd)에서 생성된 로그와 Kubernetes apiserver 및 OpenShift apiserver에서 생성되는 감사 로그입니다.



참고

내부 OpenShift Container Platform Elasticsearch 로그 저장소는 감사 로그를 위한 보안 스토리지를 제공하지 않기 때문에 감사 로그는 기본적으로 내부 Elasticsearch 인스턴스에 저장되지 않습니다. 예를 들어 Kibana에서 감사 로그를 보기 위해 감사 로그를 내부 로그 저장소로 보내려면 Forward audit logs to the log store 에 설명된 대로 로그 전달 API를 사용해야 합니다.

1.1. 클러스터 로깅 배포 정보

OpenShift Container Platform 클러스터 관리자는 OpenShift Container Platform 웹 콘솔 또는 CLI에서 클러스터 로깅을 배포하여 Elasticsearch Operator 및 Cluster Logging Operator를 설치할 수 있습니다. Operator가 설치되면 **ClusterLogging** 사용자 정의 리소스(CR)를 생성하여 클러스터 로깅 Pod 및 클러스터 로깅을 지원하는 데 필요한 기타 리소스를 예약합니다. Operator는 클러스터 로깅의 배포, 업그레이드 및 유지보수를 담당합니다.

ClusterLogging CR은 로그를 수집, 저장 및 시각화하기 위해 로깅 스택의 모든 구성 요소를 포함하는 전체 클러스터 로깅 환경을 정의합니다. Cluster Logging Operator는 클러스터 로깅 CR을 감시하고 그에 따라 로깅 배포를 조정합니다.

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

자세한 내용은 로그 수집기 구성을 참조하십시오.

1.1.1. JSON OpenShift Container Platform 로깅 정보

JSON 로깅을 사용하여 구조화된 오브젝트로 JSON 문자열을 구문 분석하도록 Log Forwarding API를 구성할 수 있습니다. 다음 작업을 수행할 수 있습니다.

- JSON 로그 구문 분석
- Elasticsearch의 JSON 로그 데이터 구성
- Elasticsearch 로그 저장소로 JSON 로그를 전달

자세한 내용은 JSON 로깅 정보를 참조하십시오.

1.1.2. Kubernetes 이벤트 수집 및 저장 정보

OpenShift Container Platform 이벤트 라우터는 Kubernetes 이벤트를 감시하고 OpenShift Container Platform 로깅에 의한 수집을 위해 해당 이벤트를 기록하는 Pod입니다. 이벤트 라우터를 수동으로 배포해야 합니다.

자세한 내용은 Kubernetes 이벤트 수집 및 저장을 참조하십시오.

1.1.3. OpenShift Container Platform Logging 업데이트 정보

OpenShift Container Platform을 사용하면 OpenShift Container Platform 로깅을 업데이트할 수 있습니다. OpenShift Container Platform Logging을 업데이트하는 동안 다음 Operator를 업데이트해야 합니다.

- Elasticsearch Operator
- Cluster Logging Operator

자세한 내용은 OpenShift Container Platform 로깅 업데이트 정보를 참조하십시오.

1.1.4. 클러스터 대시보드 보기 정보

OpenShift Container Platform 로깅 대시보드에는 클러스터 수준에서 Elasticsearch 인스턴스에 대한 세부 정보를 보여주는 차트가 포함되어 있습니다. 이 차트는 문제를 진단하고 예측하는 데 도움이 됩니다.

자세한 내용은 클러스터 대시보드 보기 정보를 참조하십시오.

1.1.5. OpenShift Container Platform 로깅 문제 해결 정보

다음 작업을 수행하여 로깅 문제를 해결할 수 있습니다.

- 로깅 상태 보기
- 로그 저장소의 상태 보기
- 로깅 경고 이해
- Red Hat 지원을 위한 로깅 데이터 수집
- 심각한 경고 문제 해결

1.1.6. OpenShift Container Platform 로깅 설치 제거 정보

ClusterLogging 사용자 정의 리소스(CR)를 삭제하여 로그 집계를 중지할 수 있습니다. CR을 삭제한 후에도 다른 클러스터 로깅 구성 요소는 남아 있으며 선택적으로 제거할 수 있습니다.

자세한 내용은 OpenShift Container Platform 로깅 설치 제거를 참조하십시오.

1.1.7. 필드 내보내기 정보

로깅 시스템 내보내기 필드. 내보낸 필드는 로그 레코드에 있으며 Elasticsearch 및 Kibana에서 검색할 수 있습니다.

자세한 내용은 필드 내보내기 정보를 참조하십시오.

1.1.8. 클러스터 로깅 구성 요소 정보

클러스터 로깅 구성 요소에는 수집기가 포함되어 있습니다. 이 수집기는 OpenShift Container Platform 클러스터의 각 노드에 배포되어 모든 노드와 컨테이너 로그를 수집한 다음 로그 저장소에 씁니다. 중앙 집중식 웹 UI에서 이렇게 집계된 데이터를 사용하여 풍부한 시각화 및 대시보드를 생성할 수 있습니다.

클러스터 로깅의 주요 구성 요소는 다음과 같습니다.

- 수집 - 클러스터에서 로그를 수집하고 형식을 지정한 후 로그 저장소로 전달하는 구성 요소입니다. 최신 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 테스트를 거쳐 단기 스토리지용으로 최적화되었습니다.
- 시각화 - 로그, 그래프, 차트 등을 보는 데 사용할 수 있는 UI 구성 요소입니다. 최신 구현은 Kibana입니다.

이 문서에서는 달리 표시된 경우를 제외하고 로그 저장소와 Elasticsearch, 시각화와 Kibana, 수집과 Fluentd를 서로 바꾸어 사용할 수 있습니다.

1.1.9. 로깅 수집기 정보

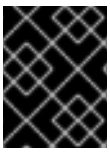
OpenShift Container Platform은 Fluentd를 사용하여 컨테이너 및 노드 로그를 수집합니다.

기본적으로 로그 수집기는 다음 소스를 사용합니다.

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

로깅 수집기는 데몬 세트로 배포되어 각 OpenShift Container Platform 노드에 Pod를 배포합니다. 시스템 및 인프라 로그는 journald가 운영 체제, 컨테이너 런타임 및 OpenShift Container Platform의 로그 메시지를 사용하여 생성합니다. 애플리케이션 로그는 CRI-O 컨테이너 엔진에 의해 생성됩니다. Fluentd는 이러한 소스에서 로그를 수집하여 OpenShift Container Platform의 구성에 따라 내부 또는 외부로 전달합니다.

컨테이너 런타임은 로그 메시지의 소스(프로젝트, Pod 이름 및 컨테이너 ID)를 식별하기 위한 최소한의 정보를 제공합니다. 이 정보로는 로그 소스를 고유하게 식별하기에 부족합니다. 로그 수집기에서 로그 처리를 시작하기 전에 지정된 이름과 프로젝트가 있는 Pod를 삭제하면 레이블 및 주석과 같은 API 서버의 정보를 사용할 수 없게 됩니다. 로그 메시지를 비슷한 이름의 Pod 및 프로젝트와 구별할 방법 또는 로그의 소스를 추적할 방법이 없을 수 있습니다. 이 제한은 로그 수집 및 정규화가 **최선의 노력**으로 간주된다는 의미입니다.



중요

사용 가능한 컨테이너 런타임은 로그 메시지의 소스를 식별할 수 있는 최소한의 정보를 제공하며, 고유한 개별 로그 메시지 또는 그러한 메시지의 소스 추적을 보장하지 않습니다.

자세한 내용은 로그 수집기 구성을 참조하십시오.

1.1.10. 로그 저장소 정보

기본적으로 OpenShift Container Platform은 ES(Elasticsearch)를 사용하여 로그 데이터를 저장합니다. 원한다면 Fluentd 프로토콜, syslog 프로토콜 또는 OpenShift Container Platform Log Forwarding API를 사용하여 로그 전송 기능으로 로그를 외부 로그 저장소로 전송할 수 있습니다.

클러스터 로깅 Elasticsearch 인스턴스는 약 7일 동안의 단기 스토리지용으로 최적화 및 테스트되었습니다. 로그를 장기간 유지하려면 데이터를 타사 스토리지 시스템으로 이동하는 것이 좋습니다.

Elasticsearch는 Fluentd의 로그 데이터를 데이터 저장소 또는 *인덱스*로 구성한 다음 각 인덱스를 *shards*라고 하는 조각 여러 개로 다시 세분화합니다. 그리고 이 조각을 Elasticsearch 클러스터의 Elasticsearch 노드 세트에 분산 배치합니다. *복제본*이라는 이름의 shard 사본을 작성하도록 Elasticsearch를 구성할 수 있습니다. Elasticsearch는 이 역시 Elasticsearch 노드에 분산 배치합니다. **ClusterLogging** 사용자 정의 리소스(CR)를 사용하면 shard의 복제 방식을 지정하여 데이터 중복성과 장애에 대한 회복 탄력성을 제공할 수 있습니다. **ClusterLogging** CR의 보존 정책을 사용하여 다양한 로그 유형의 보존 기간을 지정할 수도 있습니다.

참고

인덱스 템플릿의 기본 shard 수는 Elasticsearch 데이터 노드 수와 같습니다.

Cluster Logging Operator 및 그에 동반되는 OpenShift Elasticsearch Operator는 각 Elasticsearch 노드가 자체 스토리지 볼륨이 포함된 고유한 배포를 사용하여 배포되도록 합니다. 필요에 따라 **ClusterLogging** 사용자 정의 리소스(CR)를 사용하여 Elasticsearch 노드 수를 늘릴 수 있습니다. 스토리지 구성과 관련된 고려 사항은 Elasticsearch 설명서를 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.

자세한 내용은 로그 저장소 구성을 참조하십시오.

1.1.11. 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.

자세한 내용은 로그 시각화 프로그램 구성을 참조하십시오.

1.1.12. 이벤트 라우팅 정보

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

이벤트 라우터를 수동으로 배포해야 합니다.

자세한 내용은 Kubernetes 이벤트 수집 및 저장을 참조하십시오.

1.1.13. 로그 전송 정보

기본적으로 OpenShift Container Platform 클러스터 로깅은 **ClusterLogging** 사용자 정의 리소스(CR)에 정의된 기본 내부 Elasticsearch 로그 저장소로 로그를 보냅니다. 로그를 기타 로그 집계기로 전달하려면 로그 전달 기능을 사용하여 클러스터 내부 또는 외부의 특정 끝점으로 로그를 보내면 됩니다.

자세한 내용은 타사 시스템으로 로그 전달 을 참조하십시오.

2장. 클러스터 로깅 배포

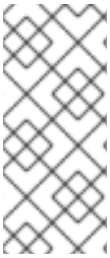
OpenShift Elasticsearch Operator 및 Cluster Logging Operator를 배포하여 클러스터 로깅을 설치할 수 있습니다. OpenShift Elasticsearch Operator는 클러스터 로깅에 사용되는 Elasticsearch 클러스터를 생성하고 관리합니다. Cluster Logging Operator는 로깅 스택의 구성 요소를 생성하고 관리합니다.

OpenShift Container Platform에 클러스터 로깅을 배포하는 프로세스에는 다음이 포함됩니다.

- 클러스터 로깅 스토리지 고려 사항 검토
- OpenShift Container Platform 웹 콘솔 또는 CLI 를 사용하여 OpenShift Elasticsearch Operator 및 Cluster Logging Operator 설치

2.1. 웹 콘솔을 사용하여 클러스터 로깅 설치

OpenShift Container Platform 웹 콘솔을 사용하여 OpenShift Elasticsearch Operator 및 Cluster Logging Operator를 설치할 수 있습니다.

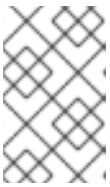


참고

기본 Elasticsearch 로그 저장소를 사용하지 않으려면 **ClusterLogging** 사용자 정의 리소스(CR)에서 내부 Elasticsearch **로그 저장소**, Kibana **시각화** 및 로그 **큐레이션** 구성 요소를 제거할 수 있습니다. 이러한 구성 요소를 제거하는 것은 선택 사항이지만 리소스를 절약할 수 있습니다. 자세한 내용은 기본 Elasticsearch 로그 저장소를 사용하지 않는 경우 사용되지 않는 구성 요소 제거를 참조하십시오.

사전 요구 사항

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



참고

영구 스토리지에 로컬 볼륨을 사용하는 경우 **LocalVolume** 개체에서 **volumeMode: block**에 설명된 원시 블록 볼륨을 사용하지 마십시오. Elasticsearch는 원시 블록 볼륨을 사용할 수 없습니다.

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 추가합니다.

절차

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

1. OpenShift Elasticsearch Operator를 설치합니다.
   a. OpenShift Container Platform 웹 콘솔에서 **Operator → OperatorHub**를 클릭합니다.
   b. 사용 가능한 Operator 목록에서 **OpenShift Elasticsearch Operator**를 선택한 다음 **설치**를 클릭합니다.

c. **설치 모드**에서 **클러스터의 모든 네임스페이스**가 선택되어 있는지 확인합니다.

d. **설치된 네임스페이스**에서 **openshift-operators-redhat**이 선택되어 있는지 확인합니다. **openshift-operators-redhat** 네임스페이스를 지정해야 합니다. **openshift-operators** 네임스페이스에 신뢰할 수 없는 Community Operator가 포함될 수 있고, 여기에서 OpenShift Container Platform 지표와 동일한 이름의 지표를 게시하면 충돌이 발생합니다.

e. **이 네임스페이스에서 Operator 권장 클러스터 모니터링 사용**을 선택합니다.
이 옵션은 네임스페이스 오브젝트에서 **openshift.io/cluster-monitoring: "true"** 레이블을 설정합니다. 클러스터 모니터링이 **openshift-operators-redhat** 네임스페이스를 스크랩하도록 하려면 이 옵션을 선택해야 합니다.

f. **4.6**을 **업데이트 채널**로 선택합니다.

g. **승인 전략**을 선택합니다.

- **자동** 전략을 사용하면 Operator 새 버전이 준비될 때 OLM(Operator Lifecycle Manager)이 자동으로 Operator를 업데이트할 수 있습니다.
- **수동** 전략을 사용하려면 적절한 자격 증명을 가진 사용자가 Operator 업데이트를 승인해야 합니다.

h. **설치**를 클릭합니다.

i. **Operator → 설치된 Operator** 페이지로 전환하여 OpenShift Elasticsearch Operator가 설치되었는지 확인합니다.

j. **상태**가 **성공**인 모든 프로젝트에 **OpenShift Elasticsearch Operator**가 나열되어 있는지 확인합니다.

2. 다음과 같이 Cluster Logging 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 확인합니다.

d. **설치된 네임스페이스**에서 **Operator 권장 네임스페이스**가 **openshift-logging**인지 확인하십시오.

e. **이 네임스페이스에서 Operator 권장 클러스터 모니터링 사용**을 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 Operator 새 버전이 준비될 때 OLM(Operator Lifecycle Manager)이 자동으로 Operator를 업데이트할 수 있습니다.
- **수동** 전략을 사용하려면 적절한 자격 증명을 가진 사용자가 Operator 업데이트를 승인해야 합니다.

h. **설치**를 클릭합니다.

i. **Operator → 설치된 Operator** 페이지로 전환하여 Cluster Logging Operator가 설치되었는지 확인합니다.

j. **클러스터 로깅**이 **openshift-logging** 프로젝트에 **성공 상태**로 나열되어 있는지 확인합니다. Operator가 설치된 것으로 나타나지 않으면 다음과 같이 추가 문제 해결을 수행합니다.

- **Operator → 설치된 Operator** 페이지로 전환하여 **상태** 열에 오류 또는 실패가 있는지 점검합니다.
- **워크로드 → Pod** 페이지로 전환하고 **openshift-logging** 프로젝트에서 문제를 보고하는 Pod의 로그를 확인합니다.

3. 다음과 같이 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 보기**를 선택합니다.

d. **ClusterLoggings** 페이지에서 **ClusterLogging 생성**을 클릭합니다.
데이터를 로드하기 위해 페이지를 새로 고쳐야 할 수도 있습니다.

e. YAML 필드에서 코드를 다음으로 교체합니다.

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 검토하십시오.

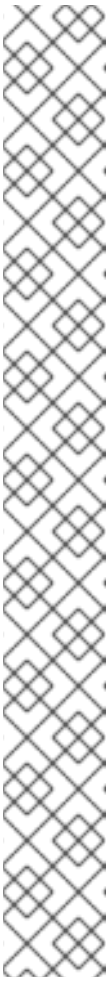
```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance" 1
  namespace: "openshift-logging"
spec:
  managementState: "Managed" 2
  logStore:
    type: "elasticsearch" 3
    retentionPolicy: 4
      application:
        maxAge: 1d
      infra:
        maxAge: 7d
      audit:
        maxAge: 7d
    elasticsearch:
      nodeCount: 3 5
      storage:
        storageClassName: "<storage-class-name>" 6
        size: 200G
      resources: 7
          limits:
            memory: "16Gi"
```

```
        requests:
          memory: "16Gi"
      proxy: 8
        resources:
          limits:
            memory: 256Mi
          requests:
            memory: 256Mi
      redundancyPolicy: "SingleRedundancy"
  visualization:
    type: "kibana" 9
    kibana:
      replicas: 1
  curation:
    type: "curator"
    curator:
      schedule: "30 3 * * *" 10
  collection:
    logs:
      type: "fluentd" 11
      fluentd: {}
```

1. 이름은 **instance**이어야 합니다.
2. 클러스터 로깅 관리 상태. 경우에 따라 클러스터 로깅 기본값을 변경하는 경우 이를 **Unmanaged**로 설정해야 합니다. 그러나 관리되지 않는 배포는 클러스터 로깅이 다시 Managed 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 스토리지 클래스 이름을 입력합니다. 최상의 성능을 위해서는 블록 스토리지를 할당하는 스토리지 클래스를 지정합니다. 스토리지 클래스를 지정하지 않으면 OpenShift Logging은 임시 스토리지를 사용합니다.
7. 필요에 따라 Elasticsearch에 대한 CPU 및 메모리 요청을 지정합니다. 이 값을 비워 두면 OpenShift Elasticsearch Operator가 대부분의 배포에 충분한 기본값으로 설정합니다. 기본값은 메모리 요청 시 **16Gi**이고 CPU 요청 시 **1**입니다.
8. 필요에 따라 Elasticsearch 프록시에 대한 CPU 및 메모리 요청을 지정합니다. 이 값을 비워 두면 OpenShift Elasticsearch Operator가 대부분의 배포에 충분한 기본값으로 설정합니다. 기본값은 메모리 요청 시 **256Mi**이고 CPU 요청 시 **100m**입니다.
9. Kibana 구성을 위한 설정입니다. CR을 사용하여 중복성을 위해 Kibana를 확장하고 Kibana 노드의 CPU 및 메모리를 구성할 수 있습니다. 자세한 내용은 **로그 시각화 프로그램 구성**을 참조하십시오.
10. Curator 일정 구성을 위한 설정입니다. Curator는 OpenShift Container Platform 4.5 이전의 Elasticsearch 인덱스 형식의 데이터를 삭제하는 데 사용되며 이후 릴리스에서 제거됩니다.

합니다.

11 Fluentd 구성을 위한 설정입니다. CR을 사용하여 Fluentd CPU 및 메모리 제한을 구성할 수 있습니다. 자세한 내용은 **Fluentd 구성**을 참조하십시오.



참고

Elasticsearch 컨트롤 플레인 노드(마스터 노드라고도 함)의 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. 컨트롤 플레인 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. **Create**를 클릭합니다. 이렇게 하면 클러스터 로깅 구성 요소, **Elasticsearch** 사용자 정의 리소스 및 구성 요소, Kibana 인터페이스가 생성됩니다.

4. 설치를 확인합니다.

   a. **워크로드 → Pod** 페이지로 전환합니다.

   b. **openshift-logging** 프로젝트를 선택합니다.
   다음 목록과 유사한 클러스터 로깅, Elasticsearch, Fluentd 및 Kibana에 대한 여러 Pod가 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

추가 리소스

- OperatorHub에서 Operator 설치

2.2. 설치 후 작업

Kibana를 사용하려면 Kibana에서 데이터를 탐색하고 시각화하기 위해 Kibana 인덱스 패턴 및 시각화를 수동으로 생성해야 합니다.

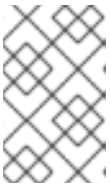
클러스터 네트워크 공급자가 네트워크 분리를 적용하는 경우 OpenShift Logging Operator가 포함된 프로젝트 간에 네트워크 트래픽을 허용합니다.

2.3. CLI를 사용하여 클러스터 로깅 설치

OpenShift Container Platform CLI를 사용하여 OpenShift Elasticsearch Operator 및 Cluster Logging Operator를 설치할 수 있습니다.

사전 요구 사항

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



참고

영구 스토리지에 로컬 볼륨을 사용하는 경우 **LocalVolume** 개체에서 **volumeMode: block**에 설명된 원시 블록 볼륨을 사용하지 마십시오. Elasticsearch는 원시 블록 볼륨을 사용할 수 없습니다.

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 추가합니다.

절차

CLI를 사용하여 OpenShift Elasticsearch Operator 및 Cluster Logging Operator를 설치하려면 다음을 수행합니다.

1. OpenShift Elasticsearch Operator의 네임스페이스를 생성합니다.
   a. OpenShift Elasticsearch Operator를 위한 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. 지표의 충돌을 방지하려면 **openshift-operators** 네임스페이스가 아니라 **openshift-operators-redhat** 네임스페이스에서 지표를 스크랩하도록 Prometheus 클러스터 모니터링 스택을 구성해야 합니다. **openshift-operators** 네임스페이스에 신뢰할 수 없는 Community Operator가 포함될 수 있고, 여기에서 OpenShift Container Platform 지표와 동일한 이름의 지표를 게시하면 충돌이 발생합니다.

(2) 문자열. 클러스터 모니터링이 **openshift-operators-redhat** 네임스페이스를 스크랩하도록 하려면 표시된 이 레이블을 지정해야 합니다.

b. 네임스페이스를 생성합니다.

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

예를 들면 다음과 같습니다.

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

2. Cluster Logging Operator의 네임스페이스를 생성합니다.

a. Cluster Logging Operator를 위한 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 생성하여 OpenShift Elasticsearch Operator를 설치합니다.

a. OpenShift Elasticsearch Operator를 위한 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 (1)
spec: {}
```

(1) **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 생성하여 네임스페이스에서 OpenShift Elasticsearch Operator를 서브스크립션합니다.

서브스크립션의 예

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: "elasticsearch-operator"
  namespace: "openshift-operators-redhat" (1)
spec:
  channel: "4.6" (2)
  installPlanApproval: "Automatic"
  source: "redhat-operators" (3)
  sourceNamespace: "openshift-marketplace"
  name: "elasticsearch-operator"
```

(1) **openshift-operators-redhat** 네임스페이스를 지정해야 합니다.

(2) **4.6**을 채널로 지정합니다.

(3) **redhat-operators**를 지정합니다. OpenShift Container Platform 클러스터가 제한된 네트워크(연결이 끊긴 클러스터)에 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
```

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

e. Operator 설치를 확인합니다.

```
$ 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    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    Elasticsearch Operator   4.6.0-202007012112.p0
Succeeded
openshift-authentication                           elasticsearch-operator.4.6.0-
202007012112.p0    Elasticsearch Operator   4.6.0-202007012112.p0
Succeeded
...
```

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

4. 다음 오브젝트를 생성하여 Cluster Logging Operator를 설치합니다.

a. Cluster Logging Operator를 위한 OperatorGroup 오브젝트 YAML 파일(예: **clo-og.yaml**)을 생성합니다.

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: cluster-logging
  namespace: openshift-logging (1)
spec:
  targetNamespaces:
  - openshift-logging (2)
```

(1) (2) **openshift-logging** 네임스페이스를 지정해야 합니다.

b. OperatorGroup 개체를 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 네임스페이스를 Cluster Logging Operator에 서브스크립션합니다.

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: cluster-logging
  namespace: openshift-logging (1)
spec:
  channel: "4.6" (2)
  name: cluster-logging
  source: redhat-operators (3)
  sourceNamespace: openshift-marketplace
```

(1) **openshift-logging** 네임스페이스를 지정해야 합니다.

(2) **4.6**을 채널로 지정합니다.

(3) **redhat-operators**를 지정합니다. OpenShift Container Platform 클러스터가 제한된 네트워크(연결이 끊긴 클러스터)에 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
```

Cluster Logging Operator는 **openshift-logging** 네임스페이스에 설치됩니다.

e. Operator 설치를 확인합니다.
openshift-logging 네임스페이스에 Cluster Logging Operator가 있어야 합니다. 버전 번호가 표시된 것과 다를 수 있습니다.

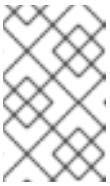
```
$ oc get csv -n openshift-logging
```

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. 다음과 같이 클러스터 로깅 인스턴스를 생성합니다.

a. Cluster Logging Operator를 위한 인스턴스 오브젝트 YAML 파일(예: **clo-instance.yaml**)을 생성합니다.



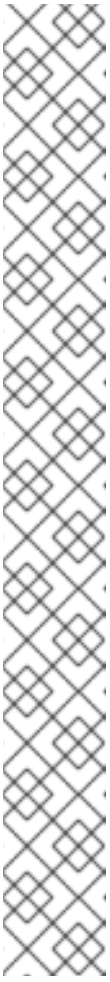
참고

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 검토하십시오.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance" 1
  namespace: "openshift-logging"
spec:
  managementState: "Managed" 2
  logStore:
    type: "elasticsearch" 3
    retentionPolicy: 4
      application:
        maxAge: 1d
      infra:
        maxAge: 7d
      audit:
        maxAge: 7d
    elasticsearch:
      nodeCount: 3 5
      storage:
        storageClassName: "<storage-class-name>" 6
        size: 200G
      resources: 7
        limits:
          memory: "16Gi"
        requests:
          memory: "16Gi"
      proxy: 8
        resources:
          limits:
            memory: 256Mi
          requests:
            memory: 256Mi
      redundancyPolicy: "SingleRedundancy"
  visualization:
    type: "kibana" 9
    kibana:
      replicas: 1
  curation:
    type: "curator"
    curator:
      schedule: "30 3 * * *" 10
  collection:
```

```
logs:
  type: "fluentd"  11
  fluentd: {}
```

1. 이름은 **instance**이어야 합니다.
2. 클러스터 로깅 관리 상태. 경우에 따라 클러스터 로깅 기본값을 변경하는 경우 이를 **Unmanaged**로 설정해야 합니다. 그러나 관리되지 않는 배포는 클러스터 로깅이 다시 Managed 상태로 될 때까지 업데이트를 받지 않습니다. 배포를 다시 Managed 상태로 설정하면 수정한 내용이 취소될 수 있습니다.
3. Elasticsearch 구성을 위한 설정입니다. CR(사용자 정의 리소스)을 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 스토리지 클래스 이름을 입력합니다. 최상의 성능을 위해서는 블록 스토리지를 할당하는 스토리지 클래스를 지정합니다. 스토리지 클래스를 지정하지 않으면 OpenShift Container Platform은 임시 스토리지로만 OpenShift Logging을 배포합니다.
7. 필요에 따라 Elasticsearch에 대한 CPU 및 메모리 요청을 지정합니다. 이러한 값을 비워 두면 OpenShift Elasticsearch Operator는 대부분의 배포에 충분한 기본값을 설정합니다. 기본값은 메모리 요청 시 **16Gi**이고 CPU 요청 시 **1**입니다.
8. 필요에 따라 Elasticsearch 프록시에 대한 CPU 및 메모리 요청을 지정합니다. 이 값을 비워 두면 OpenShift Elasticsearch Operator가 대부분의 배포에 충분한 기본값으로 설정합니다. 기본값은 메모리 요청 시 **256Mi**이고 CPU 요청 시 **100m**입니다.
9. Kibana 구성을 위한 설정입니다. CR을 사용하여 중복성을 위해 Kibana를 확장하고 Kibana 노드의 CPU 및 메모리를 구성할 수 있습니다. 자세한 내용은 **로그 시각화 프로그램 구성**을 참조하십시오.
10. Curator 일정 구성을 위한 설정입니다. Curator는 OpenShift Container Platform 4.5 이전의 Elasticsearch 인덱스 형식의 데이터를 삭제하는 데 사용되며 이후 릴리스에서 제거됩니다.
11. Fluentd 구성을 위한 설정입니다. CR을 사용하여 Fluentd CPU 및 메모리 제한을 구성할 수 있습니다. 자세한 내용은 **Fluentd 구성**을 참조하십시오.



참고

Elasticsearch 컨트롤 플레인 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. 컨트롤 플레인 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 생성합니다.

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

예를 들면 다음과 같습니다.

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

이렇게 하면 클러스터 로깅 구성 요소, **Elasticsearch** 사용자 정의 리소스 및 구성 요소, Kibana 인터페이스가 생성됩니다.

6. **openshift-logging** 프로젝트에 Pod를 나열하여 설치를 확인합니다.
다음 목록과 유사한 클러스터 로깅, Elasticsearch, Fluentd 및 Kibana에 대한 여러 Pod가 표시됩니다.

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

출력 예

```
NAME                                            READY   STATUS    RESTARTS   AGE
cluster-logging-operator-66f77ffccb-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
```

2.4. 설치 후 작업

Kibana를 사용하려면 Kibana에서 데이터를 탐색하고 시각화하기 위해 Kibana 인덱스 패턴 및 시각화를 수동으로 생성해야 합니다.

클러스터 네트워크 공급자가 네트워크 분리를 적용하는 경우 OpenShift Logging Operator가 포함된 프로젝트 간에 네트워크 트래픽을 허용합니다.

2.4.1. Kibana 인덱스 패턴 정의

인덱스 패턴은 시각화하려는 Elasticsearch 인덱스를 정의합니다. Kibana에서 데이터를 탐색하고 시각화하려면 인덱스 패턴을 생성해야 합니다.

사전 요구 사항

- Kibana에서 **인프라** 및 **감사** 인덱스를 보려면 사용자에게 **cluster-admin** 역할이나 **cluster-reader** 역할 또는 두 역할이 모두 있어야 합니다. 기본 **kubeadmin** 사용자에게는 이러한 인덱스를 나열할 수 있는 적절한 권한이 있습니다.
  default, **kube-**, **openshift-** 프로젝트에서 Pod와 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/log -n <project>
  ```

  출력 예

  ```
  yes
  ```

  참고

  감사 로그는 기본적으로 내부 OpenShift Container Platform Elasticsearch 인스턴스에 저장되지 않습니다. Kibana에서 감사 로그를 보려면 Log Forwarding API를 사용하여 감사 로그에 **default** 출력을 사용하는 파이프라인을 구성해야 합니다.

- 인덱스 패턴을 생성하려면 먼저 Elasticsearch 문서를 인덱싱해야 합니다. 이 작업은 자동으로 수행되지만 새 클러스터나 업데이트된 클러스터에서는 몇 분 정도 걸릴 수 있습니다.

프로세스

Kibana에서 인덱스 패턴을 정의하고 시각화를 생성하려면 다음을 수행합니다.

1. OpenShift Container Platform 콘솔에서 Application Launcher  를 클릭하고 **로깅**을 선택합니다.

2. **관리 → 인덱스 패턴 → 인덱스 패턴 생성**을 클릭하여 Kibana 인덱스 패턴을 생성합니다.

   - 각 사용자는 프로젝트의 로그를 보려면 Kibana에 로그인할 때 수동으로 인덱스 패턴을 생성

해야 합니다. 사용자는 **app**이라는 새 인덱스 패턴을 생성하고 **@timestamp** 시간 필드를 사용하여 컨테이너 로그를 확인해야 합니다.

- 관리자는 **@timestamp** 시간 필드를 사용하여 **app**, **infra**, **audit** 인덱스에 대해 처음 Kibana에 로그인할 때 인덱스 패턴을 생성해야 합니다.

3. 새로운 인덱스 패턴에서 Kibana 시각화를 생성합니다.

2.4.2. 네트워크 분리가 활성화될 때 프로젝트 간 트래픽 허용

클러스터 네트워크 공급자는 네트워크 분리를 실행할 수 있습니다. 이 경우 OpenShift Logging에서 배포한 operator가 포함된 프로젝트 간 네트워크 트래픽을 허용해야 합니다.

네트워크 분리는 다른 프로젝트에 있는 pod 또는 서비스 간의 네트워크 트래픽을 차단합니다. OpenShift Logging은 **openshift-operators-redhat 프로젝트에** ***OpenShift Elasticsearch Operator*** **를 설치하고 openshift- logging** 프로젝트에 *Cluster Logging Operator* 를 설치합니다. 따라서 이 두 프로젝트 간 트래픽을 허용해야 합니다.

OpenShift Container Platform은 기본 CNI(Container Network Interface) 네트워크 공급자인 OpenShift SDN과 OVN-Kubernetes에 대해 지원되는 두 가지 옵션을 제공합니다. 이 두 공급업체는 다양한 네트워크 분리 정책을 구현합니다.

OpenShift SDN에는 다음 세 가지 모드가 있습니다.

네트워크 정책

이는 기본값 모드입니다. 정책을 정의하지 않은 경우 모든 트래픽을 허용합니다. 그러나 사용자가 정책을 정의하는 경우 일반적으로 모든 트래픽을 거부한 다음 예외를 추가하여 시작합니다. 이 프로세스에서는 다른 프로젝트에서 실행 중인 애플리케이션을 중단할 수 있습니다. 따라서 하나의 로깅 관련 프로젝트에서 다른 프로젝트로 트래픽이 송신될 수 있도록 명시적으로 정책을 구성합니다.

다중 테넌트

이 모드에서는 네트워크 분리가 적용됩니다. 두 개의 로깅 관련 프로젝트에 참여하여 트래픽을 허용해야 합니다.

서브넷

이 모드에서는 모든 트래픽을 허용합니다. 네트워크 분리를 적용하지 않습니다. 아무 작업도 필요하지 않습니다.

OVN-Kubernetes는 항상 **네트워크 정책**을 사용합니다. 따라서 OpenShift SDN과 마찬가지로 하나의 로깅 관련 프로젝트에서 다른 프로젝트로 트래픽이 송신될 수 있도록 정책을 구성해야 합니다.

프로세스

- **다중 테넌트** 모드에서 OpenShift SDN을 사용하는 경우 두 프로젝트에 참여합니다. 예를 들면 다음과 같습니다.

```
$ oc adm pod-network join-projects --to=openshift-operators-redhat openshift-logging
```

- 또는 **네트워크 정책** 모드 및 OVN-Kubernetes의 OpenShift SDN의 경우 다음 작업을 수행합니다.

  a. **openshift-operators-redhat** 네임스페이스에서 레이블을 설정합니다. 예를 들면 다음과 같습니다.

```
$ oc label namespace openshift-operators-redhat project=openshift-operators-redhat
```

b. **openshift-operators-redhat** 프로젝트에서 **openshift-logging** 프로젝트로 수신할 수 있는 **openshift-logging** 네임스페이스에 네트워크 정책 오브젝트를 생성합니다. 예를 들면 다음과 같습니다.

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: allow-openshift-operators-redhat
  namespace: openshift-logging
spec:
  ingress:
    - from:
      - podSelector: {}
    - from:
      - namespaceSelector:
          matchLabels:
            project: "openshift-operators-redhat"
```

추가 리소스

- 네트워크 정책 정의
- OpenShift SDN 기본 CNI 네트워크 공급자 정보
- OVN-Kubernetes 기본 CNI(Container Network Interface) 네트워크 공급자 정보

3장. 클러스터 로깅 배포 구성

3.1. 클러스터 로깅 사용자 정의 리소스 정보

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

3.1.1. 클러스터 로깅 사용자 정의 리소스 정보

클러스터 로깅 환경을 변경하려면 **ClusterLogging** 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.

ClusterLogging 사용자 정의 리소스 (CR) 샘플

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance" 1
  namespace: "openshift-logging" 2
spec:
  managementState: "Managed" 3
  logStore:
    type: "elasticsearch" 4
    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: 5
    type: "kibana"
    kibana:
      resources:
        limits:
          memory: 736Mi
        requests:
          cpu: 100m
          memory: 736Mi
      replicas: 1
```

```
  curation: 6
    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**여야 합니다.
2. CR은 **openshift-logging** 네임스페이스에 설치해야 합니다.
3. Cluster Logging Operator 관리 상태. **Unmanaged**로 설정된 경우 Operator는 지원되지 않는 상태이며 업데이트되지 않습니다.
4. 보존 정책, 노드 수, 리소스 요청 및 제한, 스토리지 클래스를 포함한 로그 저장소 설정
5. 리소스 요청 및 제한, Pod 복제본 수를 포함한 시각화 프로그램 설정
6. 리소스 요청 및 제한, 큐레이션 스케줄 등 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 프로젝트 메타데이터로 데이터를 보강합니다.

로그 수집기의 CPU 및 메모리 제한을 구성하고 로그 수집기 Pod를 특정 노드로 이동 할 수 있습니다. **ClusterLogging** 사용자 정의 리소스(CR)의 **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 OpenShift Elasticsearch Operator 및 Cluster Logging Operator가 차이를 조정하므로 변경 사항이 사라집니다. Operator는 원래 기본적으로 모든 항목을 정의된 상태로 되돌립니다.

참고

OpenShift Container Platform 설명서에 설명되지 않은 구성이 *필요한* 경우 Cluster Logging Operator 또는 OpenShift Elasticsearch Operator를 **Unmanaged로 설정해야** *합니다*. 관리되지 않는 클러스터 로깅 환경은 *지원되지 않으며* 클러스터 로깅을 **Managed** 상태로 되돌릴 때까지 업데이트를 받지 않습니다.

3.2.2. 로깅 수집기 Pod 보기

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

프로세스

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
NODE   READINESS GATES
fluentd-8d69v  1/1    Running   0          134m    10.130.2.30   master1.example.com   <none>
<none>
fluentd-bd225  1/1    Running   0          134m    10.131.1.11   master2.example.com   <none>
<none>
fluentd-cvrzs  1/1    Running   0          134m    10.130.0.21   master3.example.com   <none>
<none>
fluentd-gpqg2  1/1    Running   0          134m    10.128.2.27   worker1.example.com   <none>
<none>
fluentd-l9j7j  1/1    Running   0          134m    10.129.2.31   worker2.example.com   <none>
<none>
```

3.2.3. 로그 수집기 CPU 및 메모리 제한 구성

로그 수집기는 CPU 및 메모리 제한을 모두 조정할 수 있습니다.

프로세스

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

```
$ oc edit ClusterLogging instance
```

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

....

spec:
  collection:
    logs:
      fluentd:
        resources:
```

```
      limits: 1
        memory: 736Mi
      requests:
        cpu: 100m
        memory: 736Mi
```

1 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 매개변수가 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 *스테이지*에 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 구성된 대로 플러시를 다시 시도합니다.

기본적으로 OpenShift Container Platform에서 Fluentd는 *지수 백오프* 방법을 사용하여 플러시를 다시 시도합니다. 여기서 Fluentd는 플러시 재시도 간격의 대기 시간을 두 배로 늘리며, 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 대기 시간을 최적화하기 위해 매개변수를 사용하여 데이터를 최대한 빨리 전송하고, 배치 누적을 방지하고, 큐와 버퍼를 더 짧게 만들고, 플러시 및 재시도를 더 자주 사용할 수 있습니다.

ClusterLogging 사용자 정의 리소스(CR)에서 다음 매개변수를 사용하여 청크 및 플러시 동작을 구성할 수 있습니다. 그러면 Fluentd에서 사용할 수 있도록 매개변수가 Fluentd 구성 맵에 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 튜닝 전용입니다. 로깅의 기능적 측면에는 영향을 미치지 않습니다.

표 3.1. 고급 Fluentd 구성 매개변수

매개변수	설명	기본
chunkLimitSize	각 청크의 최대 크기입니다. Fluentd는 이 크기에 도달하면 청크에 데이터 쓰기를 중지합니다. 그런 다음 Fluentd는 청크를 큐로 보내고 새 청크를 엽니다.	**8m**
totalLimitSize	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.	**8G**
flushInterval	청크 플러시 간격입니다. **s**(초), **m**(분), **h**(시간) 또는 **d**(일)를 사용할 수 있습니다.	**1s**
flushMode	플러시를 수행하는 방법: • **lazy**: **timekey** 매개 변수를 기반으로 청크를 플러시합니다. **timekey** 매개 변수는 수정할 수 없습니다. • **interval**: flush**Interval** **매개변수를 기반으로 청크를 플러시합니다**. • **immediate**: 데이터가 청크에 추가된 직후 청크를 플러시합니다.	**간격**
flushThreadCount	청크 플러시를 수행하는 스레드 수입니다. 스레드 수를 늘리면 플러시 처리량이 향상되어 네트워크 대기 시간이 숨겨집니다.	**2**

매개변수	설명	기본
overflowAction	큐가 가득 찼을 때 청크 동작: • **throw_exception**: 로그에 표시할 예외를 높입니다. • **block**: 전체 버퍼 문제가 해결될 때까지 데이터 청크를 중지합니다. • **drop_oldest_chunk**: 가장 오래된 청크를 삭제하여 새로 들어오는 청크를 수락합니다. 오래된 청크는 새로운 청크보다 가치가 적습니다.	**블록**
retryMaxInterval	**exponential_backoff** 재시도 방법의 최대 시간(초)입니다.	**300s**
retryType	플러시 실패 시 재시도 방법: • **exponential_backoff**: 플러시 재시도 간격을 늘립니다. Fluentd는 **retry_max_interval** 매개변수에 도달할 때까지 다음 재시도까지 대기하는 시간을 두 배로 늘립니다. • **periodic**: **retryWait 매개 변수를 기반으로 플러시를 주기적으로 재시도** 합니다.	**exponential_backoff**
retryWait	다음 청크 플러시 전의 시간(초)입니다.	**1s**

Fluentd 청크 수명 주기에 대한 자세한 내용은 Fluentd 문서의 버퍼 플러그인을 참조하십시오.

프로세스

1. **openshift-logging** 프로젝트에서 **ClusterLogging** 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 1
        flushInterval: 5s 2
        flushMode: interval 3
        flushThreadCount: 3 4
        overflowAction: throw_exception 5
        retryMaxInterval: "300s" 6
        retryType: periodic 7
        retryWait: 1s 8
        totalLimitSize: 32m 9
...
```

1. 플러시를 위해 큐에 추가되기 전에 각 청크의 최대 크기를 지정합니다.
2. 청크 플러시 간격을 지정합니다.
3. **lazy**, **interval** 또는 **immediate** 등 청크 플러시를 수행할 방법을 지정합니다.
4. 청크 플러시에 사용할 스레드 수를 지정합니다.
5. **throw_exception**, **block** 또는 **drop_oldest_chunk** 등 큐가 가득 찼을 때의 청크 동작을 지정합니다.
6. **exponential_backoff** 청크 플러시 방법의 최대 간격(초)을 지정합니다.
7. 청크 플러시 실패 시 재시도 유형을 **exponential_backoff** 또는 **periodic**으로 지정합니다.
8. 다음 청크 플러시 전 시간(초)을 지정합니다.
9. 청크 버퍼의 최대 크기를 지정합니다.

3. Fluentd Pod가 재배포되었는지 확인합니다.

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

4. 새 값이 **fluentd** 구성 맵에 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
 retry_type periodic
 retry_wait 1s
```

```
  retry_max_interval 300s
  retry_timeout 60m
  queued_chunks_limit_size "#{ENV['BUFFER_QUEUE_LIMIT'] || '32'}"
  total_limit_size 32m
  chunk_limit_size 8m
  overflow_action throw_exception
</buffer>
```

3.2.5. 기본 Elasticsearch 로그 저장소를 사용하지 않는 경우 사용되지 않은 구성 요소 제거

관리자로서 로그를 타사 로그 저장소로 전달하고 기본 Elasticsearch 로그 저장소를 사용하지 않는 경우 로깅 클러스터에서 사용하지 않는 여러 구성 요소를 제거할 수 있습니다.

즉, 기본 Elasticsearch 로그 저장소를 사용하지 않는 경우 **ClusterLogging** 사용자 정의 리소스(CR)에서 내부 Elasticsearch **logStore**, Kibana **visualization** 및 로그 **curation** 구성 요소를 제거할 수 있습니다. 이러한 구성 요소를 제거하는 것은 선택 사항이지만 리소스를 절약할 수 있습니다.

사전 요구 사항

- 로그 전달자가 로그 데이터를 기본 내부 Elasticsearch 클러스터로 전송하지 않는지 확인합니다. 로그 전달을 구성하는 데 사용한 **ClusterLogForwarder** CR YAML 파일을 검사합니다. **default**를 지정하는 **outputRefs** 요소가 *없는지* 확인합니다. 예를 들면 다음과 같습니다.

```
outputRefs:
- default
```

주의

ClusterLogForwarder CR은 로그 데이터를 내부 Elasticsearch 클러스터로 전달하고 **ClusterLogging** CR에서 **logStore** 구성 요소를 제거합니다. 이 경우 로그 데이터를 저장할 내부 Elasticsearch 클러스터가 표시되지 않습니다. 이 경우 데이터 손실이 발생할 수 있습니다.

프로세스

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

```
$ oc edit ClusterLogging instance
```

2. **ClusterLogging** CR에서 **logStore**, **visualization**, **curation** 스탠자를 제거하십시오.

3. **ClusterLogging** CR의 **collection** 스탠자를 유지합니다. 결과는 다음 예와 유사해야 합니다.

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

```
spec:
  managementState: "Managed"
  collection:
    logs:
      type: "fluentd"
      fluentd: {}
```

4. Fluentd Pod가 재배포되었는지 확인합니다.

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

추가 리소스

- 타사 시스템에 로그 전달

3.3. 로그 저장소 구성

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

다음을 포함하여 로그 저장소를 수정할 수 있습니다.

- Elasticsearch 클러스터의 스토리지
- 전체 복제에서 복제 없음까지 클러스터의 데이터 노드 간 shard 복제
- Elasticsearch 데이터에 대한 외부 액세스

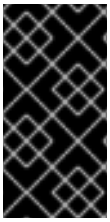
Elasticsearch는 메모리를 많이 사용하는 애플리케이션입니다. **ClusterLogging** 사용자 정의 리소스에서 달리 지정하지 않는 한 각 Elasticsearch 노드에는 메모리 요청 및 제한 모두에 16G의 메모리가 필요합니다. 초기 OpenShift Container Platform 노드 세트는 Elasticsearch 클러스터를 지원하기에 충분히 크지 않을 수 있습니다. 권장 메모리 이상으로 실행하려면 OpenShift Container Platform 클러스터에 노드를 추가해야 합니다.

각 Elasticsearch 노드는 더 낮은 메모리 설정으로 작동할 수 있지만 프로덕션 환경에는 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.

프로세스

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

1. **ClusterLogForwarder** CR YAML 파일을 생성하거나 기존 CR을 편집합니다.

   - 모든 로그 유형을 내부 Elasticsearch 인스턴스로 보내는 CR을 생성합니다. 다음 예제를 변경하지 않고 그대로 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 **ClusterLogForwarder** CR이 있는 경우 감사 로그의 기본 출력에 파이프라인을 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: http://elasticsearch-insecure.messaging.svc.cluster.local
     insecure: true
   - name: elasticsearch-secure
     type: "elasticsearch"
     url: https://elasticsearch-secure.messaging.svc.cluster.local
     secret:
       name: es-audit
   - name: secureforward-offcluster
     type: "fluentdForward"
     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 Forwading API를 사용하여 로그 전달 을 참조하십시오.

3.3.2. 로그 보존 시간 구성

기본 Elastic검색 로그 저장소가 인프라 로그, 응용 프로그램 로그 및 감사 로그의 세 가지 로그 원본 각각에 대한 인덱스를 보관하는 기간을 지정하는 *보존 정책*을 구성할 수 있습니다.

보존 정책을 구성하려면 **ClusterLogging** 사용자 정의 리소스(CR)에서 각 로그 소스에 대해 **maxAge** 매개변수를 설정합니다. CR은 Elasticsearch 롤오버 스케줄에 이러한 값을 적용하여 Elasticsearch가 롤오버된 인덱스를 삭제하는 시기를 결정합니다.

인덱스가 다음 조건 중 하나와 일치하면 Elasticsearch는 현재 인덱스를 이동하고 새 인덱스를 생성하여 인덱스를 롤오버합니다.

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

Elasticsearch는 구성한 보존 정책에 따라 롤오버된 인덱스를 삭제합니다. 로그 소스에 대한 보존 정책을 생성하지 않으면 기본적으로 7일 후에 로그가 삭제됩니다.

사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

로그 보존 시간을 구성하려면 다음을 수행합니다.

1. **retentionPolicy** 매개변수를 추가하거나 수정하려면 **ClusterLogging** 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 1일은 **1d**입니다. **maxAge**보다 오래된 로그는 삭제됩니다. 기본적으로 로그는 7일 동안 유지됩니다.

2. **Elasticsearch** 사용자 정의 리소스(CR)에서 설정을 확인할 수 있습니다.
예를 들어 Cluster Logging Operator는 8시간마다 인프라 로그의 활성 인덱스를 롤오버하는 설정이 포함된 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)
...
```

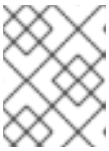
(1) 보존 정책은 각 로그 소스에 대해 해당 소스의 로그를 삭제하고 롤오버할 시기를 나타냅니다.

(2) OpenShift Container Platform이 롤오버된 인덱스를 삭제하는 경우 이 설정은 **ClusterLogging** CR에서 설정한 **maxAge**입니다.

(3) 인덱스를 롤오버할 때 고려해야 할 OpenShift Container Platform의 인덱스 수명입니다. 이

값은 **ClusterLogging** 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 **ClusterLogging** CR에서 수행해야 합니다.

OpenShift Elasticsearch Operator는 Cron 작업을 배포하고 **pollInterval**로 예약한 정의된 정책에 따라 각 매핑의 인덱스를 갱신합니다.

```
$ oc get cronjob
```

출력 예

```
NAME                  SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
curator               */10 * * * *   False     0        <none>          5s
elasticsearch-im-app     */15 * * * *   False     0        <none>          4s
elasticsearch-im-audit   */15 * * * *   False     0        <none>          4s
elasticsearch-im-infra   */15 * * * *   False     0        <none>          4s
```

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 해당 환경에 알맞은 값을 설정하므로 이러한 값을 수동으로 조정할 필요는 없습니다.

참고

대규모 클러스터에서 Elasticsearch 프록시 컨테이너의 기본 메모리 제한으로 충분하지 않을 수 있으므로 프록시 컨테이너가 OOMKilled로 됩니다. 이 문제가 발생하면 Elasticsearch 프록시에 대한 메모리 요청 및 제한을 늘립니다.

각 Elasticsearch 노드는 더 낮은 메모리 설정으로 작동할 수 있지만 프로덕션 배포에는 권장되지 **않습니다**. 프로덕션 용도의 경우 각 Pod에 기본 16Gi 이상이 할당되어 있어야 합니다. 가급적 Pod당 최대 64Gi를 할당해야 합니다.

사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

1. **openshift-logging** 프로젝트에서 **ClusterLogging** 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: 1
         limits:
           memory: "16Gi"
         requests:
           cpu: "1"
           memory: "16Gi"
       proxy: 2
         resources:
           limits:
             memory: 100Mi
           requests:
             memory: 100Mi
```

1 필요에 따라 Elasticsearch에 대한 CPU 및 메모리 요청을 지정합니다. 이 값을 비워 두면 OpenShift Elasticsearch Operator가 대부분의 배포에 충분한 기본값으로 설정합니다. 기본값은 메모리 요청 시 **16Gi**이고 CPU 요청 시 **1**입니다.

2 필요에 따라 Elasticsearch 프록시에 대한 CPU 및 메모리 요청을 지정합니다. 이 값을 비워 두면 OpenShift Elasticsearch Operator가 대부분의 배포에 충분한 기본값으로 설정합니다. 기본값은 메모리 요청 시 **256Mi**이고 CPU 요청 시 **100m**입니다.

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"
```

쿠버네티스는 일반적으로 노드 구성을 준수하며 Elasticsearch가 지정된 제한을 사용하도록 허용하지 않습니다. **requests** 및 **limits**에 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.

프로세스

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

```
$ oc edit clusterlogging instance
```

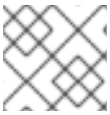
```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"

....

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. 이 방법은 안전성이 가장 높지만 필요한 디스크 양이 가장 많고 성능이 가장 낮습니다.
- **MultipleRedundancy**. Elasticsearch는 각 인덱스의 기본 shard를 데이터 노드의 절반으로 완전히 복제합니다. 이 방법은 안전성과 성능 사이의 균형이 우수합니다.
- **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 실패하는 경우 로그를 사용할 수 없거나 로그가 손실될 수 있습니다. 안전보다 성능이 더 중요하거나 자체 디스크/PVC 백업/복원 전략을 구현한 경우 이 모드를 사용합니다.



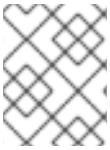
참고

인덱스 템플릿의 기본 shard 수는 Elasticsearch 데이터 노드 수와 같습니다.

3.3.5. Elasticsearch Pod 축소

클러스터에서 Elasticsearch Pod 수를 줄이면 데이터 손실 또는 Elasticsearch 성능 저하가 발생할 수 있습니다.

축소하는 경우 Pod를 한 번에 하나씩 축소하고 클러스터에서 shard와 복제본의 균형을 다시 조정할 수 있어야 합니다. Elasticsearch 상태가 **green**으로 돌아가면 다른 Pod에서 축소할 수 있습니다.



참고

Elasticsearch 클러스터가 **ZeroRedundancy**로 설정된 경우 Elasticsearch Pod를 축소해서는 안 됩니다.

3.3.6. 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.

사전 요구 사항

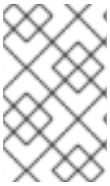
- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

1. **ClusterLogging** 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 예에서는 클러스터의 각 데이터 노드가 AWS General Purpose SSD(gp2) 스토리지 "200G"를 요청하는 영구 볼륨 클레임에 바인딩되도록 지정합니다.



참고

영구 스토리지에 로컬 볼륨을 사용하는 경우 **LocalVolume** 개체에서 **volumeMode: block**에 설명된 원시 블록 볼륨을 사용하지 마십시오. Elasticsearch는 원시 블록 볼륨을 사용할 수 없습니다.

3.3.7. emptyDir 스토리지에 대한 로그 저장소 구성

emptyDir을 로그 저장소와 함께 사용하면 임시 배포가 생성되고 재시작 시 Pod의 모든 데이터가 손실됩니다.

참고

emptyDir을 사용할 때 로그 스토리지가 다시 시작되거나 재배포되면 데이터가 손실됩니다.

사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

1. emptyDir을 지정하려면 **ClusterLogging** CR을 편집합니다.

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

3.3.8. Elasticsearch 롤링 클러스터 재시작 수행

elasticsearch 구성 맵 또는 **elasticsearch-*** 배포 구성을 변경할 때 롤링 재시작을 수행합니다.

또한 Elasticsearch Pod가 실행되는 노드를 재부팅해야 하는 경우에도 롤링 재시작이 권장됩니다.

사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

클러스터를 롤링 재시작하려면 다음을 수행합니다.

1. **openshift-loggin** 프로젝트로 변경합니다.

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

2. Elasticsearch pod의 이름을 가져옵니다.

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

3. Fluentd Pod를 축소하여 Elasticsearch로 새 로그 전송을 중지합니다.

```
$ oc -n openshift-logging patch daemonset/logging-fluentd -p '{"spec":{"template":{"spec":
{"nodeSelector":{"logging-infra-fluentd": "false"}}}}}'
```

4. OpenShift Container Platform **es_util** 툴을 사용하여 shard 동기화 플러시를 수행하여 종료하기 전에 디스크에 쓰기 대기 중인 작업이 없는지 확인하십시오.

```
$ oc exec <any_es_pod_in_the_cluster> -c elasticsearch -- es_util --query="_flush/synced" -
XPOST
```

예를 들면 다음과 같습니다.

```
$ oc exec -c elasticsearch-cdm-5ceex6ts-1-dcd6c4c7c-jpw6  -c elasticsearch -- es_util --
query="_flush/synced" -XPOST
```

출력 예

```
{"_shards":{"total":4,"successful":4,"failed":0},".security":
{"total":2,"successful":2,"failed":0},".kibana_1":{"total":2,"successful":2,"failed":0}}
```

5. 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 -d '{ "persistent": { "cluster.routing.allocation.enable" :
"primaries" } }'
```

예를 들면 다음과 같습니다.

```
$ oc exec elasticsearch-cdm-5ceex6ts-1-dcd6c4c7c-jpw6 -c elasticsearch -- es_util --
query="_cluster/settings" -XPUT -d '{ "persistent": { "cluster.routing.allocation.enable" :
"primaries" } }'
```

출력 예

```
{"acknowledged":true,"persistent":{"cluster":{"routing":{"allocation":
{"enable":"primaries"}}}},"transient":
```

6. 명령이 완료되면 ES 클러스터의 각 배포에 대해 다음을 수행합니다.

   a. 기본적으로 OpenShift Container Platform Elasticsearch 클러스터는 노드에 대한 롤아웃을 차단합니다. 다음 명령을 사용하여 롤아웃을 허용하고 Pod가 변경 사항을 선택하도록 합니다.

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

   예를 들면 다음과 같습니다.

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

   출력 예

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

   새 Pod가 배포되었습니다. Pod에 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. 배포가 완료되면 롤아웃을 허용하지 않도록 Pod를 재설정합니다.

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

예를 들면 다음과 같습니다.

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

출력 예

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

c. Elasticsearch 클러스터가 **green** 또는 **yellow** 상태인지 확인하십시오.

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

참고

이전 명령에서 사용한 Elasticsearch Pod에서 롤아웃을 수행한 경우 그 Pod는 더 이상 존재하지 않으며 여기에 새 Pod 이름이 필요합니다.

예를 들면 다음과 같습니다.

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

```
{
  "cluster_name" : "elasticsearch",
  "status" : "yellow", 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 매개변수 값이 **green** 또는 **yellow**인지 확인하십시오.

7. Elasticsearch ConfigMap을 변경한 경우 각 Elasticsearch Pod에 대해 이 단계를 반복합니다.

8. 클러스터의 모든 배포가 롤아웃되면 shard 밸런싱을 다시 활성화합니다.

```
$ oc exec <any_es_pod_in_the_cluster> -c elasticsearch -- es_util --
query="_cluster/settings" -XPUT -d '{ "persistent": { "cluster.routing.allocation.enable" : "all" }
}'
```

예를 들면 다음과 같습니다.

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

출력 예

```
{
  "acknowledged" : true,
  "persistent" : { },
  "transient" : {
    "cluster" : {
      "routing" : {
        "allocation" : {
          "enable" : "all"
        }
      }
    }
  }
}
```

9. Fluentd Pod를 확장하여 Elasticsearch에 새 로그를 전송합니다.

```
$ oc -n openshift-logging patch daemonset/logging-fluentd -p '{"spec":{"template":{"spec":
{"nodeSelector":{"logging-infra-fluentd": "true"}}}}}'
```

3.3.9. 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 생성하여 로그 저장소에 액세스할 수 있습니다. 그런 후 다음을 포함하는 cURL 요청으로 로그 저장소 서비스를 호스팅하는 노드에 액세스합니다.

- **인증: 전달자 ${token}**
- Elasticsearch 재암호화 경로 및 Elasticsearch API 요청

내부에서는 다음 명령 중 하나로 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.

프로세스

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** 매개변수를 설정할 필요는 없습니다.

b. 다음 명령을 실행하여 이전 단계에서 생성한 경로 YAML에 로그 저장소 CA 인증서를 추가합니다.

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

c. 경로를 생성합니다.

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

출력 예

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

4. Elasticsearch 서비스가 노출되어 있는지 확인합니다.

a. 요청에 사용할 이 서비스 계정의 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}"
```

응답은 다음과 유사하게 나타납니다.

출력 예

```
{
  "name" : "elasticsearch-cdm-i40ktba0-1",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "0eY-tJzcR3KOdpgeMJo-MQ",
  "version" : {
  "number" : "6.8.1",
  "build_flavor" : "oss",
  "build_type" : "zip",
  "build_hash" : "Unknown",
  "build_date" : "Unknown",
  "build_snapshot" : true,
```

```
  "lucene_version" : "7.7.0",
  "minimum_wire_compatibility_version" : "5.6.0",
  "minimum_index_compatibility_version" : "5.0.0"
},
  "<tagline>" : "<for search>"
}
```

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.

프로세스

1. **openshift-logging** 프로젝트에서 **ClusterLogging** 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.

3.4.2. 로그 시각화 프로그램 노드의 확장성 중복

중복성에 대해 로그 시각화 프로그램을 호스팅하는 Pod를 확장할 수 있습니다.

프로세스

1. **openshift-logging** 프로젝트에서 **ClusterLogging** 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)
```

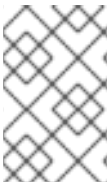
(1) Kibana 노드의 수를 지정합니다.

3.5. 클러스터 로깅 스토리지 구성

Elasticsearch는 메모리를 많이 사용하는 애플리케이션입니다. 기본 클러스터 로깅 설치는 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 환경에는 권장되지 않습니다.

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

각 Elasticsearch 배포 구성에는 영구 볼륨이 필요합니다. OpenShift Container Platform에서는 영구 볼륨 클레임을 사용합니다.



참고

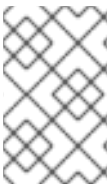
영구 스토리지에 로컬 볼륨을 사용하는 경우 **LocalVolume** 개체에서 **volumeMode: block**에 설명된 원시 블록 볼륨을 사용하지 마십시오. Elasticsearch는 원시 블록 볼륨을 사용할 수 없습니다.

OpenShift 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.

Elasticsearch에는 대규모 병합 작업을 수행하기 위해 충분한 메모리가 필요합니다. 메모리가 충분하지 않으면 응답하지 않습니다. 이 문제를 방지하려면 애플리케이션 로그 데이터 양을 계산하고 사용 가능한 스토리지 용량의 약 2배를 할당합니다.

기본적으로 스토리지 용량이 85%인 경우 Elasticsearch는 새 데이터를 노드에 할당하는 것을 중지합니다. 90%에서 Elasticsearch는 가능한 경우 기존 shard를 해당 노드에서 다른 노드로 재배치합니다. 그러나 사용 가능한 용량이 85% 미만일 때 노드에 여유 스토리지 공간이 없는 경우 Elasticsearch는 새 인덱스 생성을 거부하고 RED가 됩니다.



참고

이 낮은 워터마크 값과 높은 워터마크 값은 현재 릴리스에서 Elasticsearch 기본값입니다. 이러한 기본값을 수정할 수 있습니다. 경고가 동일한 기본값을 사용하지만 경고에서 이러한 값을 변경할 수 없습니다.

3.5.2. 추가 리소스

- 영구 Elasticsearch 스토리지

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.

프로세스

1. **openshift-logging** 프로젝트에서 **ClusterLogging** 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. 허용 오차를 사용하여 클러스터 로깅 POD 배치 제어

taint와 허용 오차를 사용하여 클러스터 로깅 Pod가 특정 노드에서 실행되고 해당 노드에서 다른 워크로드가 실행되지 않도록 할 수 있습니다.

taint와 허용 오차는 간단한 **키:값** 쌍입니다. 노드의 taint는 해당 taint를 허용하지 않는 모든 Pod를 거절하도록 노드에 지시합니다.

key는 최대 253자의 문자열이고 **value**은 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 Pod에 추가됩니다.

2 이 허용 오차는 Kibana Pod에 추가됩니다.

3 이 허용 오차는 로깅 수집기 Pod에 추가됩니다.

3.7.1. 허용 오차를 사용하여 로그 저장소 Pod 배치 제어

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

ClusterLogging 사용자 정의 리소스(CR)를 통해 로그 저장소 Pod에 허용 오차를 적용하고 노드 사양을 통해 노드에 taint를 적용합니다. 노드의 taint는 해당 taint를 허용하지 않는 모든 Pod를 거절하도록 노드에 지시하는 **key:value pair**입니다. 다른 Pod에 없는 특정 **key:value** 쌍을 사용하는 경우 해당 노드에서는 로그 저장소 Pod만 실행할 수 있습니다.

기본적으로 로그 저장소 Pod에는 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.

프로세스

1. 다음 명령을 사용하여 클러스터 로깅 Pod를 예약하려는 노드에 taint를 추가합니다.

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

   예를 들면 다음과 같습니다.

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

   이 예에서는 키 **elasticsearch**, 값 **node** 및 taint 효과 **NoExecute**로 **node1**에 taint를 배치합니다. **NoExecute** 효과가 있는 노드는 taint와 일치하는 Pod만 스케줄링하고 일치하지 않는 기존 Pod는 제거합니다.

2. Elasticsearch Pod에 대한 허용 오차를 구성하려면 **ClusterLogging** CR의 **logstore** 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 **elasticsearch** 키의 taint가 존재할 것을 요구하도록 **Exists** Operator를 지정합니다.
   3. **NoExecute** 효과를 지정합니다.
   4. 선택적으로 **tolerationSeconds** 매개변수를 지정하여 Pod가 제거되기 전까지 노드에 바인딩되는 시간을 설정합니다.

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

3.7.2. 허용 오차를 사용하여 로그 시각화 프로그램 Pod 배치 제어

Pod의 허용 오차를 사용하여 로그 시각화 프로그램 Pod가 실행되는 노드를 제어하고 다른 워크로드가 해당 노드를 사용하지 못하게 할 수 있습니다.

ClusterLogging 사용자 정의 리소스(CR)를 통해 로그 시각화 프로그램 Pod에 허용 오차를 적용하고 노드 사양을 통해 노드에 taint를 적용합니다. 노드의 taint는 해당 taint를 허용하지 않는 모든 Pod를 거절하도록 노드에 지시하는 **key:value pair**입니다. 다른 Pod에 없는 특정 **key:value** 쌍을 사용하는 경우 해당 노드에서는 Kibana Pod만 실행할 수 있습니다.

사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

1. 다음 명령을 사용하여 로그 시각화 프로그램 Pod를 예약하려는 노드에 taint를 추가합니다.

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

예를 들면 다음과 같습니다.

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

이 예에서는 키 **kibana**, 값 **node** 및 taint 효과 **NoExecute**로 **node1**에 taint를 배치합니다. **NoExecute** taint 효과를 사용해야 합니다. **NoExecute**는 taint와 일치하는 Pod만 스케줄링하고 일치하지 않는 기존 Pod는 제거합니다.

2. Kibana Pod에 대한 허용 오차를 구성하려면 **ClusterLogging** CR의 **visualization** 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. **key/value/effect** 매개변수가 일치할 것을 요구하도록 **Exists** Operator를 지정합니다.
3. **NoExecute** 효과를 지정합니다.
4. 선택적으로 **tolerationSeconds** 매개변수를 지정하여 Pod가 제거되기 전까지 노드에 바인딩되는 시간을 설정합니다.

이 허용 오차는 **oc adm taint** 명령으로 생성된 taint와 일치합니다. 이 허용 오차가 있는 Pod는 **node1**에 스케줄링할 수 있습니다.

3.7.3. 허용 오차를 사용하여 로그 수집기 Pod 배치 제어

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

ClusterLogging 사용자 정의 리소스(CR)를 통해 로깅 수집기 Pod에 허용 오차를 적용하고 노드 사양을 통해 노드에 taint를 적용합니다. taint 및 허용 오차를 사용하여 메모리나 CPU 문제 등으로 인해 Pod가 제거되지 않도록 할 수 있습니다.

기본적으로 로깅 수집기 Pod에는 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.

프로세스

1. 다음 명령을 사용하여 로깅 수집기 Pod에서 로깅 수집기 Pod를 스케줄링할 노드에 taint를 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 **collector**, 값 **node** 및 taint 효과 **NoExecute**로 **node1**에 taint를 배치합니다. **NoExecute** taint 효과를 사용해야 합니다. **NoExecute**는 taint와 일치하는 Pod만 스케줄링하고 일치하지 않는 기존 Pod는 제거합니다.

2. **ClusterLogging** 사용자 정의 리소스(CR)의 **collection** 스탠자를 편집하여 로깅 수집기 Pod에 대한 허용 오차를 구성합니다.

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

   (1) 노드에 추가한 키를 지정합니다.

2. **key/value/effect** 매개변수가 일치할 것을 요구하도록 **Exists** Operator를 지정합니다.
3. **NoExecute** 효과를 지정합니다.
4. 선택적으로 **tolerationSeconds** 매개변수를 지정하여 Pod가 제거되기 전까지 노드에 바인딩되는 시간을 설정합니다.

이 허용 오차는 **oc adm taint** 명령으로 생성된 taint와 일치합니다. 이 허용 오차가 있는 Pod는 **node1**에 스케줄링할 수 있습니다.

3.7.4. 추가 리소스

- taint 및 허용 오차에 대한 자세한 내용은 노드 taint를 사용하여 Pod 배치 제어 를 참조하십시오.

3.8. 노드 선택기로 클러스터 로깅 리소스 이동

노드 선택기를 사용하여 Elasticsearch, Kibana, Curator Pod를 다른 노드에 배포할 수 있습니다.

3.8.1. 클러스터 로깅 리소스 이동

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

예를 들어 높은 CPU, 메모리 및 디스크 요구 사항으로 인해 Elasticsearch Pod를 다른 노드로 옮길 수 있습니다.

사전 요구 사항

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

프로세스

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

```
$ oc edit ClusterLogging instance
```

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

...

spec:
  collection:
    logs:
      fluentd:
        resources: null
      type: fluentd
  curation:
    curator:
      nodeSelector: 1
        node-role.kubernetes.io/infra: ''
```

```
      resources: null
      schedule: 30 3 * * *
    type: curator
  logStore:
    elasticsearch:
      nodeCount: 3
      nodeSelector: 2
        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: 3
        node-role.kubernetes.io/infra: ''
      proxy:
        resources: null
      replicas: 1
      resources: null
    type: kibana

...
```

1 2 3 적절한 값이 설정된 **nodeSelector** 매개변수를 이동하려는 구성 요소에 추가합니다. 표시된 형식으로 **nodeSelector**를 사용하거나 노드에 지정된 값에 따라 **<key>: <value>** 쌍을 사용할 수 있습니다.

검증

oc get pod -o wide 명령을 사용하여 구성 요소가 이동했는지 확인할 수 있습니다.

예를 들면 다음과 같습니다.

- **ip-10-0-147-79.us-east-2.compute.internal** 노드에서 Kibana pod를 이동하려고 경우 다음을 실행합니다.

```
$ 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 Pod를 전용 인프라 노드인 **ip-10-0-139-48.us-east-2.compute.internal** 노드로 이동하려는 경우 다음을 실행합니다.

```
$ 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 있음에 유의합니다.

```
$ 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 pod를 이동하려면 **ClusterLogging** CR을 편집하여 노드 선택기를 추가합니다.

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

...

spec:

...

  visualization:
    kibana:
      nodeSelector: 1
        node-role.kubernetes.io/infra: ''
      proxy:
        resources: null
      replicas: 1
      resources: null
    type: kibana
```

```
-
```

1 노드 사양의 레이블과 일치하는 노드 선택기를 추가합니다.

- CR을 저장하면 현재 Kibana pod가 종료되고 새 pod가 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 pod는 **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 pod가 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 **RateLimitIntervalSec=30s** 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.

프로세스

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

```
Compress=yes (1)
ForwardToConsole=no (2)
ForwardToSyslog=no
MaxRetentionSec=1month (3)
RateLimitBurst=10000 (4)
RateLimitIntervalSec=30s
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 입력합니다. 기본값은 **1month**입니다.

(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 **RateLimitIntervalSec=30s** 및 **RateLimitBurst=10000**을 설정하는 것이 좋습니다.

5 로그 저장 방법을 지정합니다. 기본값은 **persistent**입니다.

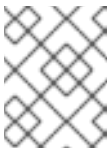
- **/var/log/journal/**에서 메모리에 로그를 저장하기 위한 **volatile**입니다.
- **/var/log/journal/**의 디스크에 로그를 저장하기 위한 **persistent**입니다. systemd는 디렉토리가 없는 경우 디렉토리를 생성합니다.
- 디렉토리가 존재하는 경우 **/var/log/journal/**에 로그를 저장하기 위한 **auto**입니다. 존재하지 않는 경우 systemd는 **/run/systemd/journal**에 로그를 임시 저장합니다.
- 로그를 저장하지 않는 **none**입니다. 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.html을 참조하십시오. 해당 페이지에 나열된 기본 설정은 OpenShift Container Platform에 적용되지 않을 수 있습니다.

2. **journal.conf** 파일을 base64로 변환하고 다음 명령을 실행하여 **jrnl_cnf** 라는 변수에 저장합니다.

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

3. 이전 단계에서 만든 **jrnl_cnf** 변수가 포함된 **MachineConfig** 오브젝트를 만듭니다. 다음 샘플 명령은 작업자에 대한 **MachineConfig** 오브젝트를 생성합니다.

```
$ cat << EOF > ./40-worker-custom-journald.yaml 1
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker 2
  name: 40-worker-custom-journald 3
spec:
  config:
    ignition:
      config: {}
```

```
    security:
      tls: {}
    timeouts: {}
    version: 3.1.0
  networkd: {}
  passwd: {}
  storage:
    files:
    - contents:
        source: data:text/plain;charset=utf-8;base64,${jrnl_cnf} 4
        verification: {}
      filesystem: root
      mode: 0644 5
      path: /etc/systemd/journald.conf.d/custom.conf
 osImageURL: ""
EOF
```

1. 선택 사항: 컨트롤 플레인 (마스터) 노드의 경우 파일 이름을 **40-master-custom-journald.yaml** 로 지정할 수 있습니다.
2. 선택 사항: 컨트롤 플레인 (마스터) 노드의 경우 역할을 **master** 로 제공합니다.
3. 선택 사항: 컨트롤 플레인 (마스터) 노드의 경우 이름을 **40-master-custom-journald** 로 지정할 수 있습니다.
4. 선택 사항: the **journald.conf** 파일에 매개 변수의 정적 사본을 포함하려면 **${jrnl_cnf}** 를 **echo $jrnl_cnf** 명령의 출력으로 바꿉니다.
5. **journal.conf** 파일에 대한 권한을 설정합니다. **0644** 권한을 설정하는 것이 좋습니다.

4. 머신 구성을 생성합니다.

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

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

5. 각 노드에 새로 렌더링된 구성의 롤아웃 상태를 모니터링합니다.

```
$ oc describe machineconfigpool/<node> 1
```

1. 노드를 **master** 또는 **worker** 로 지정합니다.

작업자의 출력 예

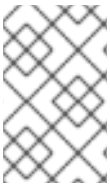
```
Name:         worker
Namespace:
Labels:       machineconfiguration.openshift.io/mco-built-in=
Annotations:  <none>
API Version:  machineconfiguration.openshift.io/v1
Kind:         MachineConfigPool

...
```

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

3.10. 로그 큐레이터 구성

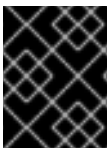
로그 보존 시간을 구성할 수 있습니다. 인프라 로그, 애플리케이션 로그 및 감사 로그의 세 가지 로그 소스 각각에 대해 별도의 보존 정책을 사용하여 기본 Elasticsearch 로그 저장소가 인덱스를 유지하는 기간을 지정할 수 있습니다. 자세한 내용은 로그 보존 시간 구성 을 참조하십시오.



참고

로그 데이터를 큐레이션하려면 로그 보존 시간을 구성하는 것이 좋습니다. OpenShift Container Platform 4.4 및 이전의 현재 데이터 모델과 이전 데이터 모델 모두에서 작동합니다.

필요한 경우 OpenShift Container Platform 4.4 및 이전 버전에서 데이터 모델을 사용하는 Elasticsearch 인덱스를 제거하려면 Elasticsearch Curator를 사용할 수도 있습니다. 다음 섹션에서는 Elasticsearch Curator 사용 방법에 대해 설명합니다.



중요

Elasticsearch Curator는 OpenShift Container Platform 4.7 (OpenShift Logging 5.0)에서 더 이상 사용되지 않으며 OpenShift Logging 5.1에서 제거됩니다.

3.10.1. Curator 일정 구성

OpenShift Logging 설치로 생성된 **ClusterLogging** 사용자 정의 리소스를 사용하여 Curator 일정을 지정할 수 있습니다.

중요

Elasticsearch Curator는 OpenShift Container Platform 4.7 (OpenShift Logging 5.0)에서 더 이상 사용되지 않으며 OpenShift Logging 5.1에서 제거됩니다.

사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

Curator 일정을 구성하려면 다음을 수행합니다.

1. **openshift-logging** 프로젝트에서 **ClusterLogging** 사용자 정의 리소스를 편집합니다.

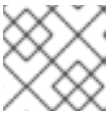
```
$ oc edit clusterlogging instance
```

```
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 Pod가 실행되는 호스트 노드를 기준으로 설정됩니다.

3.10.2. Curator 인덱스 삭제 구성

OpenShift Container Platform 버전 4.5 이전의 데이터 모델을 사용하는 Elasticsearch 데이터를 삭제하도록 Elasticsearch Curator를 구성할 수 있습니다. 프로젝트별 및 전역 설정을 구성할 수 있습니다. 전역 설정은 지정되지 않은 모든 프로젝트에 적용됩니다. 프로젝트별 설정은 전역 설정보다 우선합니다.

중요

Elasticsearch Curator는 OpenShift Container Platform 4.7 (OpenShift Logging 5.0)에서 더 이상 사용되지 않으며 OpenShift Logging 5.1에서 제거됩니다.

사전 요구 사항

- 클러스터 로깅이 설치되어 있어야 합니다.

프로세스

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 매개변수는 다음과 같습니다.

표 3.2. 프로젝트 옵션

변수 이름	설명

변수 이름	설명
project_name	myapp-devel과 같은 프로젝트의 실제 이름입니다. OpenShift Container Platform **작업** 로그의 경우 프로젝트 이름으로 이름 **.operations**를 사용합니다.
작업	수행할 작업으로, 현재 **삭제**만 허용됩니다.
단위	삭제에 사용할 기간으로,**days**, **weeks**, 또는 **months**이 있습니다.
값	단위 수입니다.

표 3.3. 필터 옵션

변수 이름	설명
.defaults	**.defaults**를 **project_name**으로 사용하여 지정되지 않은 프로젝트의 기본값을 설정합니다.
.regex	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 **1 day**이 지난 인덱스 삭제
- **1 week**가 지난 **myapp-qe** 프로젝트에서 인덱스 삭제
- **8 weeks**가 지난 **작업** 로그 삭제
- **31 days**이 지난 후 다른 모든 프로젝트 인덱스 삭제
- **^project\..+\-dev.*$** regex와 일치하는 1일이 지난 인덱스 삭제
- **^project\..+\-test.*$** regex와 일치하는 2일이 지난 인덱스 삭제

다음을 사용하십시오.

```
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 **months**을 사용하면 Curator는 현재 달의 현재 날짜가 아니라 현재 달의 첫 날부터 계산을 시작합니다. 예를 들어 오늘이 4월 15일이고 오늘보다 2개월 지난 인덱스를 삭제하려는 경우(삭제: 개월: 2) Curator는 2월 15일 이전의 인덱스를 삭제하지 않습니다. 2월 1일 이전의 인덱스를 삭제합니다. 즉, 현재 달의 첫날로 되돌아간 다음 해당 날짜로부터 총 두 달 전으로 되돌아갑니다. Curator를 정확하게 사용하려면 일(예: **삭제: days)을 사용하는 것이 좋습니다. 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 OpenShift Elasticsearch Operator 및 Cluster Logging Operator가 차이를 조정하므로 변경 사항이 사라집니다. Operator는 원래 기본적으로 모든 항목을 정의된 상태로 되돌립니다.

참고

OpenShift Container Platform 설명서에 설명되지 않은 구성이 *필요한* 경우 Cluster Logging Operator 또는 OpenShift Elasticsearch Operator를 **Unmanaged로 설정해야** *합니다*. 관리되지 않는 클러스터 로깅 환경은 *지원되지 않으며* 클러스터 로깅을 **Managed** 상태로 되돌릴 때까지 업데이트를 받지 않습니다.

3.11.2. 지원되지 않는 로깅 구성

다음 구성 요소를 수정하려면 Cluster Logging Operator를 Unmanaged 상태로 설정해야 합니다.

- Curator cron 작업
- **Elasticsearch** CR
- Kibana 배포
- **fluent.conf** 파일
- Fluentd 데몬 세트

다음 구성 요소를 수정하려면 OpenShift Elasticsearch Operator를 Unmanaged 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 구성 맵을 사용하여 Curator 작업 파일을 수정할 수 없습니다.

3.11.3. 관리되지 않는 Operator에 대한 지원 정책

Operator의 *관리 상태*는 Operator가 설계 의도에 따라 클러스터의 해당 구성 요소에 대한 리소스를 적극적으로 관리하고 있는지 여부를 판별합니다. *Unmanaged* 상태로 설정된 Operator는 구성 변경에 응답하지 않고 업데이트되지도 않습니다.

비프로덕션 클러스터 또는 디버깅 중에는 이 기능이 유용할 수 있지만, Unmanaged 상태의 Operator는 지원되지 않으며 개별 구성 요소의 구성 및 업그레이드를 클러스터 관리자가 전적으로 통제하게 됩니다.

다음과 같은 방법으로 Operator를 Unmanaged 상태로 설정할 수 있습니다.

- **개별 Operator 구성**
  개별 Operator는 구성에 **managementState** 매개변수가 있습니다. Operator에 따라 다양한 방식으로 이 매개변수에 액세스할 수 있습니다. 예를 들어, Cluster Logging Operator는 관리 대상인 사용자 정의 리소스(CR)를 수정하여 이를 수행하는 반면 Cluster Samples Operator는 클러스터 전체의 구성 리소스를 사용합니다.

  managementState 매개변수를 **Unmanaged**로 변경하면 Operator가 리소스를 적극적으로 관리하지 않으며 해당하는 구성 요소와 관련된 조치도 수행하지 않습니다. 클러스터가 손상되고 수동 복구가 필요할 가능성이 있으므로 이 관리 상태를 지원하지 않는 Operator도 있습니다.

주의

개별 Operator를 **Unmanaged** 상태로 변경하면 특정 구성 요소 및 기능이 지원되지 않습니다. 지원을 계속하려면 보고된 문제를 **Managed** 상태에서 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 재정의를 설정하면 전체 클러스터가 지원되지 않는 상태가 됩니다. 지원을 계속하려면 재정의를 제거한 후 보고된 문제를 재현해야 합니다.

4장. 리소스의 로그 보기

OpenShift CLI(oc) 및 웹 콘솔을 사용하여 빌드, 배포 및 Pod와 같은 다양한 리소스의 로그를 볼 수 있습니다.

참고

리소스 로그는 제한된 로그 보기 기능을 제공하는 기본 기능입니다. 로그 검색 및 보기 환경을 개선하려면 OpenShift Container Platform 클러스터 로깅 을 설치하는 것이 좋습니다. 클러스터 로깅은 노드 시스템 감사 로그, 애플리케이션 컨테이너 로그 및 인프라 로그와 같은 OpenShift Container Platform 클러스터의 모든 로그를 전용 로그 저장소로 집계합니다. 그런 다음 Kibana 인터페이스를 통해 로그 데이터를 쿼리, 검색 및 시각화할 수 있습니다. 리소스 로그는 클러스터 로깅 로그 저장소에 액세스하지 않습니다.

4.1. 리소스 로그 보기

OpenShift CLI(oc) 및 웹 콘솔에서 다양한 리소스의 로그를 볼 수 있습니다. 로그는 로그의 말미 또는 끝에서 읽습니다.

사전 요구 사항

- OpenShift CLI(oc)에 액세스합니다.

프로세스(UI)

1. OpenShift Container Platform 콘솔에서 **워크로드 → Pod**로 이동하거나 조사하려는 리소스를 통해 Pod로 이동합니다.

참고

빌드와 같은 일부 리소스에는 직접 쿼리할 Pod가 없습니다. 이러한 인스턴스에서 리소스의 **세부 정보** 페이지에서 **로그** 링크를 찾을 수 있습니다.

2. 드롭다운 메뉴에서 프로젝트를 선택합니다.
3. 조사할 Pod 이름을 클릭합니다.
4. **로그**를 클릭합니다.

프로세스(CLI)

- 특정 Pod의 로그를 확인합니다.

```
$ oc logs -f <pod_name> -c <container_name>
```

다음과 같습니다.

-f

선택 사항: 출력이 로그에 기록되는 내용을 따르도록 지정합니다.

<pod_name>

pod 이름을 지정합니다.

<container_name>

선택 사항: 컨테이너의 이름을 지정합니다. Pod에 여러 컨테이너가 있는 경우 컨테이너 이름을 지정해야 합니다.

예를 들면 다음과 같습니다.

```
$ oc logs ruby-58cd97df55-mww7r
```

```
$ oc logs -f ruby-57f7f4855b-znl92 -c ruby
```

로그 파일의 내용이 출력됩니다.

- 특정 리소스의 로그를 확인합니다.

```
$ oc logs <object_type>/<resource_name> 1
```

1 리소스 유형 및 이름을 지정합니다.

예를 들면 다음과 같습니다.

```
$ oc logs deployment/ruby
```

로그 파일의 내용이 출력됩니다.

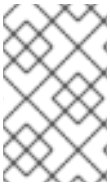
5장. KIBANA를 사용하여 클러스터 로그 보기

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 사용하면 데이터로 다음을 수행할 수 있습니다.

- **검색** 탭을 사용하여 데이터를 검색하고 찾습니다.
- **시각화** 탭을 사용하여 데이터를 차트로 작성하고 매핑합니다.
- **대시보드** 탭을 사용하여 사용자 정의 대시보드를 생성하고 봅니다.

Kibana 인터페이스의 사용 및 구성은 이 문서의 범위를 벗어납니다. 인터페이스 사용에 대한 자세한 내용은 Kibana 문서를 참조하십시오.



참고

감사 로그는 기본적으로 내부 OpenShift Container Platform Elasticsearch 인스턴스에 저장되지 않습니다. Kibana에서 감사 로그를 보려면 Log Forwarding API를 사용하여 감사 로그에 **default** 출력을 사용하는 파이프라인을 구성해야 합니다.

5.1. KIBANA 인덱스 패턴 정의

인덱스 패턴은 시각화하려는 Elasticsearch 인덱스를 정의합니다. Kibana에서 데이터를 탐색하고 시각화하려면 인덱스 패턴을 생성해야 합니다.

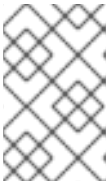
사전 요구 사항

- Kibana에서 **인프라** 및 **감사** 인덱스를 보려면 사용자에게 **cluster-admin** 역할이나 **cluster-reader** 역할 또는 두 역할이 모두 있어야 합니다. 기본 **kubeadmin** 사용자에게는 이러한 인덱스를 나열할 수 있는 적절한 권한이 있습니다.
  default, **kube-**, **openshift-** 프로젝트에서 Pod와 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/log -n <project>
  ```

  출력 예

  ```
  yes
  ```

  참고

  감사 로그는 기본적으로 내부 OpenShift Container Platform Elasticsearch 인스턴스에 저장되지 않습니다. Kibana에서 감사 로그를 보려면 Log Forwarding API를 사용하여 감사 로그에 **default** 출력을 사용하는 파이프라인을 구성해야 합니다.

- 인덱스 패턴을 생성하려면 먼저 Elasticsearch 문서를 인덱싱해야 합니다. 이 작업은 자동으로 수행되지만 새 클러스터나 업데이트된 클러스터에서는 몇 분 정도 걸릴 수 있습니다.

프로세스

Kibana에서 인덱스 패턴을 정의하고 시각화를 생성하려면 다음을 수행합니다.

1. OpenShift Container Platform 콘솔에서 Application Launcher 를 클릭하고 **로깅**을 선택합니다.
2. **관리 → 인덱스 패턴 → 인덱스 패턴 생성**을 클릭하여 Kibana 인덱스 패턴을 생성합니다.
   - 각 사용자는 프로젝트의 로그를 보려면 Kibana에 로그인할 때 수동으로 인덱스 패턴을 생성해야 합니다. 사용자는 **app**이라는 새 인덱스 패턴을 생성하고 **@timestamp** 시간 필드를 사용하여 컨테이너 로그를 확인해야 합니다.
   - 관리자는 **@timestamp** 시간 필드를 사용하여 **app**, **infra**, **audit** 인덱스에 대해 처음 Kibana에 로그인할 때 인덱스 패턴을 생성해야 합니다.
3. 새로운 인덱스 패턴에서 Kibana 시각화를 생성합니다.

5.2. KIBANA에서 클러스터 로그 보기

Kibana 웹 콘솔에서 클러스터 로그를 봅니다. 이 문서의 범위를 벗어난 Kibana에서 데이터를 보고 시각화하는 방법입니다. 자세한 내용은 Kibana 설명서를 참조하십시오.

사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.
- Kibana 인덱스 패턴이 있어야 합니다.
- Kibana에서 **인프라** 및 **감사** 인덱스를 보려면 사용자에게 **cluster-admin** 역할이나 **cluster-reader** 역할 또는 두 역할이 모두 있어야 합니다. 기본 **kubeadmin** 사용자에게는 이러한 인덱스를 나열할 수 있는 적절한 권한이 있습니다.
  default, **kube-**, **openshift-** 프로젝트에서 Pod와 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/log -n <project>
```

출력 예

```
yes
```

참고

감사 로그는 기본적으로 내부 OpenShift Container Platform Elasticsearch 인스턴스에 저장되지 않습니다. Kibana에서 감사 로그를 보려면 Log Forwarding API를 사용하여 감사 로그에 **default** 출력을 사용하는 파이프라인을 구성해야 합니다.

프로세스

Kibana에서 로그를 보려면 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.

3. Kibana에서 **검색**을 클릭합니다.

4. 왼쪽 상단 드롭다운 메뉴에서 생성한 인덱스 패턴(**app**, **audit** 또는 **infra**)을 선택합니다.
   로그 데이터가 타임스탬프가 있는 문서로 표시됩니다.

5. 타임스탬프가 있는 문서 중 하나를 확장합니다.

6. **JSON** 탭을 클릭하여 해당 문서에 대한 로그 항목을 표시합니다.

예 5.1. Kibana의 샘플 인프라 로그 항목


```
{
  "_index": "infra-000001",
  "_type": "_doc",
  "_id": "YmJmYTBlNDkZTRmLTliMGQtMjE3NmFiOGUyOWM3",
  "_version": 1,
  "_score": null,
  "_source": {
    "docker": {
      "container_id": "f85fa55bbef7bb783f041066be1e7c267a6b88c4603dfce213e32c1"
    },
    "kubernetes": {
      "container_name": "registry-server",
      "namespace_name": "openshift-marketplace",
      "pod_name": "redhat-marketplace-n64gc",
      "container_image": "registry.redhat.io/redhat/redhat-marketplace-index:v4.6",
      "container_image_id": "registry.redhat.io/redhat/redhat-marketplace-
index@sha256:65fc0c45aabb95809e376feb065771ecda9e5e59cc8b3024c4545c168f",
      "pod_id": "8f594ea2-c866-4b5c-a1c8-a50756704b2a",
      "host": "ip-10-0-182-28.us-east-2.compute.internal",
      "master_url": "https://kubernetes.default.svc",
      "namespace_id": "3abab127-7669-4eb3-b9ef-44c04ad68d38",
      "namespace_labels": {
        "openshift_io/cluster-monitoring": "true"
      },
      "flat_labels": [
        "catalogsource_operators_coreos_com/update=redhat-marketplace"
      ]
    },
    "message": "time=\"2020-09-23T20:47:03Z\" level=info msg=\"serving registry\"
database=/database/index.db port=50051",
    "level": "unknown",
    "hostname": "ip-10-0-182-28.internal",
    "pipeline_metadata": {
      "collector": {
        "ipaddr4": "10.0.182.28",
        "inputname": "fluent-plugin-systemd",
        "name": "fluentd",
        "received_at": "2020-09-23T20:47:15.007583+00:00",
        "version": "1.7.4 1.6.0"
      }
    },
    "@timestamp": "2020-09-23T20:47:03.422465+00:00",
```

```
    "viaq_msg_id": "YmJmYTBlNDktMDMGQtMjE3NmFiOGUyOWM3",
    "openshift": {
      "labels": {
        "logging": "infra"
      }
    }
  },
  "fields": {
    "@timestamp": [
      "2020-09-23T20:47:03.422Z"
    ],
    "pipeline_metadata.collector.received_at": [
      "2020-09-23T20:47:15.007Z"
    ]
  },
  "sort": [
    1600894023422
  ]
}
```

6장. 타사 시스템에 로그 전달

기본적으로 클러스터 로깅은 컨테이너 및 인프라 로그를 **ClusterLogging** 사용자 정의 리소스에 정의된 기본 내부 Elasticsearch 로그 저장소로 보냅니다. 그러나 보안 스토리지를 제공하지 않기 때문에 감사 로그를 내부 저장소로 보내지 않습니다. 이 기본 구성이 요구 사항을 충족하는 경우 Log Forwarding API를 구성할 필요가 없습니다.

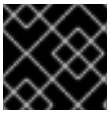
다른 로그 집계기에 로그를 보내려면 OpenShift Container Platform Log Forwarding API를 사용합니다. 이 API를 사용하면 컨테이너, 인프라 및 감사 로그를 클러스터 내부 또는 외부의 특정 엔드포인트에 보낼 수 있습니다. 다른 유형의 로그를 다양한 시스템에 보낼 수 있으므로 각 유형에 서로 다른 사용자가 액세스할 수 있습니다. 또한 조직의 필요에 따라 로그를 안전하게 보낼 수 있도록 TLS 지원을 활성화할 수도 있습니다.

참고

내부 로그 저장소로 감사 로그를 보내려면 로그 저장소에 감사 로그 전달 의 설명에 따라 Log Forwarding API를 사용하십시오.

로그를 외부로 전달할 때 Cluster Logging Operator는 Fluentd 구성 맵을 생성하거나 수정하여 원하는 프로토콜을 통해 로그를 보냅니다. 외부 로그 집계기에서 프로토콜을 구성해야 합니다.

또는 구성 맵을 생성하고 Fluentd **forward** 프로토콜 또는 syslog 프로토콜을 사용하여 외부 시스템으로 로그를 전송할 수 있습니다. 그러나 이러한 로그 전달 방법은 OpenShift Container Platform에서 더 이상 사용되지 않으며 향후 릴리스에서 제거됩니다.



중요

동일한 클러스터에서 구성 맵 방법과 Log Forwarding API를 사용할 수 없습니다.

6.1. 타사 시스템으로 로그 전달 정보

클러스터 로그를 외부의 타사 시스템으로 전달하려면 OpenShift Container Platform 클러스터 내부 및 외부의 특정 끝점으로 로그를 전송하기 위해 **ClusterLogForwarder** 사용자 정의 리소스(CR)에 지정된 *출력*과 *파이프라인*의 조합을 사용해야 합니다. *입력*을 사용하여 특정 프로젝트와 관련된 애플리케이션 로그를 끝점으로 전달할 수도 있습니다.

- *출력*은 사용자가 정의한 로그 데이터의 대상 또는 로그를 보낼 위치입니다. 출력은 다음 유형 중 하나일 수 있습니다.
  - **elasticsearch**. 외부 Elasticsearch 6(모든 릴리스) 인스턴스입니다. **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 시크릿에 **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.
  - **default**. 내부 OpenShift Container Platform Elasticsearch 인스턴스입니다. 기본 출력을 구성할 필요는 없습니다. **default** 출력을 구성하는 경우 **default** 출력이 Cluster Logging Operator용으로 예약되므로 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.
  - **application**. 인프라 컨테이너 애플리케이션을 제외하고 클러스터에서 실행 중인 사용자 애플리케이션에 의해 생성된 컨테이너 로그입니다.
  - **infrastructure**. **openshift***, **kube*** 또는 **default** 프로젝트에서 실행되는 Pod의 컨테이너 로그 및 노드 파일 시스템에서 가져온 저널 로그입니다.
  - **audit**. auditd에서 생성된 로그, 노드 감사 시스템, Kubernetes API 서버 및 OpenShift API 서버의 감사 로그입니다.

  파이프라인에서 **key:value** 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 **default** 출력이 있는 파이프라인이 없으면 로그가 기본 Elasticsearch 인스턴스로 전달되지 않습니다.
- 기본적으로 클러스터 로깅은 컨테이너 및 인프라 로그를 **ClusterLogging** 사용자 정의 리소스에 정의된 기본 내부 Elasticsearch 로그 저장소로 보냅니다. 그러나 보안 스토리지를 제공하지 않기 때문에 감사 로그를 내부 저장소로 보내지 않습니다. 이 기본 구성이 요구 사항을 충족하는 경우 Log Forwarding API를 구성하지 마십시오.
- 로그 유형에 대한 파이프라인을 정의하지 않으면 정의되지 않은 유형의 로그가 삭제됩니다. 예를 들어 **application** 및 **audit** 유형에 대한 파이프라인을 지정하고 **infrastructure** 유형에 대한 파이프라인을 지정하지 않으면 **infrastructure** 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 보호되도록 하는 것이 좋습니다. OpenShift Container Platform 클러스터 로깅은 이러한 규정을 준수하지 않습니다.
- 키 및 시크릿, 서비스 계정, 포트 열기 또는 전역 프록시 구성과 같이 외부 대상에 필요할 수 있는 추가 구성을 생성하고 유지보수할 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 (1)
  namespace: openshift-logging (2)
spec:
  outputs:
   - name: elasticsearch-secure (3)
     type: "elasticsearch"
     url: https://elasticsearch.secure.com:9200
     secret:
       name: elasticsearch
   - name: elasticsearch-insecure (4)
     type: "elasticsearch"
     url: http://elasticsearch.insecure.com:9200
   - name: kafka-app (5)
     type: "kafka"
     url: tls://kafka.secure.com:9093/app-topic
  inputs: (6)
   - name: my-app-logs
     application:
       namespaces:
       - my-project
  pipelines:
   - name: audit-logs (7)
     inputRefs:
      - audit
     outputRefs:
      - elasticsearch-secure
      - default
     labels:
       secure: "true" (8)
       datacenter: "east"
   - name: infrastructure-logs (9)
     inputRefs:
      - infrastructure
     outputRefs:
      - elasticsearch-insecure
     labels:
       datacenter: "west"
   - name: my-app (10)
     inputRefs:
      - my-app-logs
     outputRefs:
      - default
   - inputRefs: (11)
      - application
     outputRefs:
      - kafka-app
     labels:
       datacenter: "south"
```

(1) **ClusterLogForwarder** CR의 이름은 **instance**여야 합니다.

2 **ClusterLogForwarder** CR의 네임스페이스는 **openshift-logging**이어야 합니다.

3 보안 시크릿과 보안 URL을 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 끝점에서 요구하는 시크릿입니다. **openshift-logging** 프로젝트에 이 시크릿이 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-project** 네임스페이스에서 애플리케이션 로그를 필터링하기 위한 입력 구성입니다.

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

- 선택 사항입니다. 파이프라인을 설명하는 이름입니다.
- **inputRefs**는 로그 유형이며 이 예에서는 **audit**입니다.
- **outputRefs**는 사용할 출력의 이름입니다.이 예에서 **elasticsearch-secure**는 보안 Elasticsearch 인스턴스로 전달하고 **default**은 내부 Elasticsearch 인스턴스로 전달합니다.
- 선택 사항: 로그에 추가할 레이블입니다.

8 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다. "true"와 같은 인용 값은 부울 값이 아닌 문자열 값으로 인식됩니다.

9 인프라 로그를 안전하지 않은 외부 Elasticsearch 인스턴스로 전송하는 파이프라인 구성:

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

- 선택 사항입니다. 파이프라인을 설명하는 이름입니다.
- **inputRefs**는 특정 입력인 **my-app-logs**입니다.
- **outputRefs**는 **default**입니다.
- 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

11 파이프라인 이름 없이 Kafka 브로커에 로그를 전송하는 파이프라인 구성:

- **inputRefs**는 이 예제 **application**에서 로그 유형입니다.
- **outputRefs**는 사용할 출력의 이름입니다.
- 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

외부 로그 집계기를 사용할 수 없는 경우 Fluentd 로그 처리

외부 로깅 집계기를 사용할 수 없으며 로그를 수신할 수 없는 경우 Fluentd는 계속 로그를 수집하여 버퍼에 저장합니다. 로그 집계기를 사용할 수 있게 되면 버퍼링된 로그를 포함하여 로그 전달이 재개됩니다. 버퍼가 완전히 채워지면 Fluentd는 로그 수집을 중지합니다. OpenShift Container Platform은 로그를 회전시켜 삭제합니다. Fluentd 데몬 세트 또는 pod에 버퍼 크기를 조정하거나 PVC(영구 볼륨 클레임)를 추가할 수 없습니다.

6.1.1. 외부 Elasticsearch 인스턴스로 로그 전달

선택적으로 내부 OpenShift Container Platform Elasticsearch 인스턴스에 추가하거나 대신 외부 Elasticsearch 인스턴스에 로그를 전달할 수 있습니다. OpenShift Container Platform에서 로그 데이터를 수신하도록 외부 로그 집계기를 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 **default** 출력을 사용하여 내부 인스턴스로 로그를 전달하는 파이프라인을 생성합니다. **default** 출력을 생성할 필요가 없습니다. **default** 출력을 구성하는 경우 **default** 출력이 Cluster Logging Operator용으로 예약되므로 오류 메시지가 나타납니다.

참고

내부 OpenShift Container Platform Elasticsearch 인스턴스**에만** 로그를 전달하려는 경우 **ClusterLogForwarder** CR을 생성할 필요가 없습니다.

사전 요구 사항

- 지정된 프로토콜 또는 형식을 사용하여 로깅 데이터를 수신하도록 구성된 로깅 서버가 있어야 합니다.

절차

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:
      myLabel: "myValue" 11
  - name: infrastructure-audit-logs 12
    inputRefs:
    - infrastructure
    outputRefs:
    - elasticsearch-insecure
    labels:
      logs: "audit-infra"
```

1. **ClusterLogForwarder** CR의 이름은 **instance**여야 합니다.
2. **ClusterLogForwarder** 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 끝점에서 요구하는 시크릿 이름을 지정해야 합니다. 시크릿은 **openshift-logging** 프로젝트에 있어야 하며 각각의 인증서를 가리키는 **tls.crt**, **tls.key** 및 **ca-bundle.crt** 키가 있어야 합니다.
7. 선택 사항: 파이프라인의 이름을 지정합니다.
8. **application, infrastructure**, 또는 **audit**등 해당 파이프라인을 사용하여 전달해야 하는 로그 유형을 지정합니다.
9. 로그 전달을 위해 해당 파이프라인과 함께 사용할 출력을 지정합니다.
10. 선택 사항: 로그를 내부 Elasticsearch 인스턴스로 보내려면 **기본** 출력을 지정합니다.
11. 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.
12. 선택 사항: 지원되는 유형의 다른 외부 로그 집계기로 로그를 전달하도록 다중 출력을 구성합니다.

    - 선택 사항: 파이프라인을 설명하는 이름입니다.

- **inputRefs**는 **application, infrastructure** 또는 **audit** 등 해당 파이프라인을 사용하여 전달할 로그 유형입니다.
- **outputRefs**는 사용할 출력의 이름입니다.
- 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

2. CR 오브젝트를 생성합니다.

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

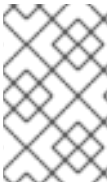
Cluster Logging Operator는 Fluentd Pod를 재배포합니다. Pod가 재배포되지 않으면 Fluentd Pod를 삭제하여 강제로 재배포할 수 있습니다.

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

6.1.2. Fluentd 정방향 프로토콜을 사용하여 로그 전달

Fluentd **전달** 프로토콜을 사용하여 프로토콜을 수락하도록 구성된 외부 로그 집계기로 로그 사본을 보낼 수 있습니다. 기본 Elasticsearch 로그 저장소를 사용하여 추가하거나 대신 이 작업을 수행할 수 있습니다. OpenShift Container Platform에서 로그 데이터를 수신하도록 외부 로그 수집기를 구성해야 합니다.

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



참고

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

사전 요구 사항

- 지정된 프로토콜 또는 형식을 사용하여 로깅 데이터를 수신하도록 구성된 로깅 서버가 있어야 합니다.

절차

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. **ClusterLogForwarder** 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 끝점에서 요구하는 시크릿 이름을 지정해야 합니다. 시크릿은 **openshift-logging** 프로젝트에 있어야 하며 각각의 인증서를 가리키는 **tls.crt**, **tls.key** 및 **ca-bundle.crt** 키가 있어야 합니다.
7. 선택 사항입니다. 파이프라인의 이름을 지정합니다.
8. **application, infrastructure**, 또는 **audit**등 해당 파이프라인을 사용하여 전달해야 하는 로그 유형을 지정합니다.
9. 로그 전달을 위해 해당 파이프라인과 함께 사용할 출력을 지정합니다.
10. 선택 사항입니다. 로그를 내부 Elasticsearch 인스턴스로 전달하려면 **default** 출력을 지정합니다.
11. 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.
12. 선택 사항: 지원되는 유형의 다른 외부 로그 집계기로 로그를 전달하도록 다중 출력을 구성합니다.

    - 선택 사항: 파이프라인을 설명하는 이름입니다.

- **inputRefs**는 **application, infrastructure** 또는 **audit** 등 해당 파이프라인을 사용하여 전달할 로그 유형입니다.
- **outputRefs**는 사용할 출력의 이름입니다.
- 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

2. CR 오브젝트를 생성합니다.

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

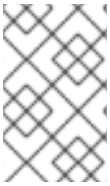
Cluster Logging Operator는 Fluentd Pod를 재배포합니다. Pod가 재배포되지 않으면 Fluentd Pod를 삭제하여 강제로 재배포할 수 있습니다.

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

6.1.3. syslog 프로토콜을 사용하여 로그 전달

syslog RFC3164 또는 RFC5424 프로토콜을 사용하여 기본 Elasticsearch 로그 저장소 대신 또는 기본 Elasticsearch 로그 저장소에 더하여 해당 프로토콜을 수락하도록 구성된 외부 로그 집계기에 로그 사본을 보낼 수 있습니다. OpenShift Container Platform에서 로그를 수신하도록 syslog 서버와 같은 외부 로그 수집기를 구성해야 합니다.

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



참고

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

사전 요구 사항

- 지정된 프로토콜 또는 형식을 사용하여 로깅 데이터를 수신하도록 구성된 로깅 서버가 있어야 합니다.

절차

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: rsyslog-east 3
     type: syslog 4
     syslog: 5
       facility: local0
```

```
      rfc: RFC3164
      payloadKey: message
      severity: informational
    url: 'tls://rsyslogserver.east.example.com:514' 6
    secret: 7
      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 8
    inputRefs: 9
    - audit
    - application
    outputRefs: 10
    - rsyslog-east
    - default 11
    labels:
      secure: "true" 12
      syslog: "east"
  - name: syslog-west 13
    inputRefs:
    - infrastructure
    outputRefs:
    - rsyslog-west
    - default
    labels:
      syslog: "west"
```

1. **ClusterLogForwarder** CR의 이름은 **instance**여야 합니다.
2. **ClusterLogForwarder** CR의 네임스페이스는 **openshift-logging**이어야 합니다.
3. 출력 이름을 지정합니다.
4. **syslog** 유형을 지정합니다.
5. 선택 사항입니다. 아래 나열된 syslog 매개변수를 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 끝점에서 요구하는 시크릿 이름을 지정해야 합니다. 시크릿은 **openshift-logging** 프로젝트에 있어야 하며 각각의 인증서를 가리키는 **tls.crt**, **tls.key** 및 **ca-bundle.crt** 키가 있어야 합니다.
8. 선택 사항: 파이프라인의 이름을 지정합니다.

9 **application, infrastructure**, 또는 **audit**등 해당 파이프라인을 사용하여 전달해야 하는 로그 유형을 지정합니다.

10 로그 전달을 위해 해당 파이프라인과 함께 사용할 출력을 지정합니다.

11 선택 사항: 로그를 내부 Elasticsearch 인스턴스로 전달하려면 **default** 출력을 지정합니다.

12 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다. "true"와 같은 인용 값은 부울 값이 아닌 문자열 값으로 인식됩니다.

13 선택 사항: 지원되는 유형의 다른 외부 로그 집계기로 로그를 전달하도록 다중 출력을 구성합니다.

- 선택 사항: 파이프라인을 설명하는 이름입니다.
- **inputRefs**는 **application, infrastructure** 또는 **audit** 등 해당 파이프라인을 사용하여 전달할 로그 유형입니다.
- **outputRefs**는 사용할 출력의 이름입니다.
- 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

2. CR 오브젝트를 생성합니다.

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

Cluster Logging Operator는 Fluentd Pod를 재배포합니다. Pod가 재배포되지 않으면 Fluentd Pod를 삭제하여 강제로 재배포할 수 있습니다.

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

6.1.3.1. Syslog 매개변수

syslog 출력에 대해 다음을 구성할 수 있습니다. 자세한 내용은 syslog RFC3164 또는 RFC5424 RFC를 참조하십시오.

- 기능: syslog 기능. 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 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**

- Optional. **payloadKey**: syslog 메시지의 페이로드로 사용할 레코드 필드입니다.

참고

payloadKey 매개변수를 구성하면 다른 매개 변수가 syslog로 전달되지 않습니다.

- rfc: syslog를 사용하여 로그를 보내는 데 사용할 RFC입니다. 기본값은 RFC5424입니다.
- 심각도: 발신 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: tag는 syslog 메시지에서 태그로 사용할 레코드 필드를 지정합니다.
- trimPrefix: 태그에서 지정된 접두사를 제거합니다.

6.1.3.2. 추가 RFC5424 syslog 매개변수

RFC5424에는 다음 매개변수가 적용됩니다.

- appName: APP-NAME은 로그를 전송한 애플리케이션을 식별하는 자유 텍스트 문자열입니다. **RFC5424**에 대해 지정해야 합니다.
- msgID: MSGID는 메시지 유형을 식별하는 자유 텍스트 문자열입니다. **RFC5424**에 대해 지정해야 합니다.

- procID: PROCID는 자유 텍스트 문자열입니다. 값이 변경되면 syslog 보고가 중단되었음을 나타냅니다. **RFC5424**에 대해 지정해야 합니다.

6.1.4. 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 이 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.

프로세스

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: tcp://kafka.devlab2.example.com:9093/infra-topic 7
   - name: audit-logs
     type: kafka
     url: tls://kafka.qelab.example.com:9093/audit-topic
     secret:
        name: kafka-secret-qe
  pipelines:
   - name: app-topic 8
     inputRefs: 9
     - application
     outputRefs: 10
     - app-logs
     labels:
       logType: "application" 11
   - name: infra-topic 12
     inputRefs:
     - infrastructure
     outputRefs:
     - infra-logs
     labels:
       logType: "infra"
   - name: audit-topic
     inputRefs:
     - audit
```

```
  outputRefs:
  - audit-logs
  - default 13
  labels:
    logType: "audit"
```

1. **ClusterLogForwarder** CR의 이름은 **instance**여야 합니다.
2. **ClusterLogForwarder** 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 끝점에서 요구하는 시크릿 이름을 지정해야 합니다. 시크릿은 **openshift-logging** 프로젝트에 있어야 하며 각각의 인증서를 가리키는 **tls.crt**, **tls.key** 및 **ca-bundle.crt** 키가 있어야 합니다.
7. 선택 사항: 안전하지 않은 출력을 보내려면 URL 앞에 있는 **tcp** 접두사를 사용합니다. 이 출력에서 **secret** 키와 해당 **name**을 생략합니다.
8. 선택 사항: 파이프라인의 이름을 지정합니다.
9. **application, infrastructure**, 또는 **audit**등 해당 파이프라인을 사용하여 전달해야 하는 로그 유형을 지정합니다.
10. 로그 전달을 위해 해당 파이프라인과 함께 사용할 출력을 지정합니다.
11. 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.
12. 선택 사항: 지원되는 유형의 다른 외부 로그 집계기로 로그를 전달하도록 다중 출력을 구성합니다.
    - 선택 사항: 파이프라인을 설명하는 이름입니다.
    - **inputRefs**는 **application, infrastructure** 또는 **audit** 등 해당 파이프라인을 사용하여 전달할 로그 유형입니다.
    - **outputRefs**는 사용할 출력의 이름입니다.
    - 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.
13. 선택 사항: 내부 Elasticsearch 인스턴스로 로그를 전달하려면 **기본값** 을 지정합니다.

2. 선택 사항: 단일 출력을 여러 kafka 브로커로 전달하려면 다음 예와 같이 kafka 브로커 배열을 지정합니다.

```
...
spec:
  outputs:
  - name: app-logs
```

```
    type: kafka
    secret:
      name: kafka-secret-dev
    kafka: 1
      brokers: 2
        - tls://kafka-broker1.example.com:9093/
        - tls://kafka-broker2.example.com:9093/
      topic: app-topic 3
...
```

1. **brokers** 및 **topic** 키가 있는 **kafka** 키를 지정합니다.
2. **brokers** 키를 사용하여 하나 이상의 브로커 배열을 지정합니다.
3. **topic** 키를 사용하여 로그를 수신할 대상 항목을 지정합니다.

3. CR 오브젝트를 생성합니다.

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

Cluster Logging Operator는 Fluentd Pod를 재배포합니다. Pod가 재배포되지 않으면 Fluentd Pod를 삭제하여 강제로 재배포할 수 있습니다.

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

6.1.5. 특정 프로젝트의 애플리케이션 로그 전달

Cluster Log Forwarder를 사용하여 특정 프로젝트의 애플리케이션 로그 사본을 외부 로그 수집기로 보낼 수 있습니다. 기본 Elasticsearch 로그 저장소를 사용하여 추가하거나 대신 이 작업을 수행할 수 있습니다. OpenShift Container Platform에서 로그 데이터를 수신하도록 외부 로그 수집기를 구성해야 합니다.

프로젝트의 애플리케이션 로그 전달을 구성하려면 프로젝트에서 하나 이상의 입력, 다른 로그 집계기에 대한 선택적 출력, 이러한 입력 및 출력을 사용하는 파이프라인을 사용하여 **ClusterLogForwarder** 사용자 정의 리소스(CR)를 생성합니다.

사전 요구 사항

- 지정된 프로토콜 또는 형식을 사용하여 로깅 데이터를 수신하도록 구성된 로깅 서버가 있어야 합니다.

절차

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'
inputs: 7
  - name: my-app-logs
    application:
      namespaces:
      - my-project
pipelines:
  - name: forward-to-fluentd-insecure 8
    inputRefs: 9
    - my-app-logs
    outputRefs: 10
    - fluentd-server-insecure
    labels: 11
      project: "my-project"
  - name: forward-to-fluentd-secure 12
    inputRefs:
    - application
    - audit
    - infrastructure
    outputRefs:
    - fluentd-server-secure
    - default
    labels:
      clusterId: "C1234"
```

1. **ClusterLogForwarder** CR의 이름은 **instance**여야 합니다.
2. **ClusterLogForwarder** CR의 네임스페이스는 **openshift-logging**이어야 합니다.
3. 출력 이름을 지정합니다.
4. 출력 유형을 **elasticsearch**, **fluentdForward**, **syslog** 또는 **kafka**로 지정합니다.
5. 외부 로그 집계기의 URL 및 포트를 유효한 절대 URL로 지정합니다. 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 끝점에서 요구하는 시크릿 이름을 지정해야 합니다. 시크릿은 **openshift-logging** 프로젝트에 있어야 하며 각각의 인증서를 가리키는 **tls.crt**, **tls.key** 및 **ca-bundle.crt** 키가 있어야 합니다.
7. 지정된 프로젝트에서 애플리케이션 로그를 필터링하기 위한 입력 구성입니다.
8. 입력을 사용하여 프로젝트 애플리케이션 로그를 외부 Fluentd 인스턴스로 보내는 파이프라인 구성입니다.
9. **my-app-logs** 입력입니다.
10. 사용할 출력의 이름입니다.

11 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

12 로그를 다른 로그 집계기로 보내는 파이프라인 구성입니다.

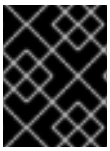
- 선택 사항: 파이프라인의 이름을 지정합니다.
- **application, infrastructure**, 또는 **audit**등 해당 파이프라인을 사용하여 전달해야 하는 로그 유형을 지정합니다.
- 로그 전달을 위해 해당 파이프라인과 함께 사용할 출력을 지정합니다.
- 선택 사항: 로그를 내부 Elasticsearch 인스턴스로 전달하려면 **default** 출력을 지정합니다.
- 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

2. CR 오브젝트를 생성합니다.

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

6.1.6. 레거시 Fluentd 방법을 사용하여 로그 전달

Fluentd **전달** 프로토콜을 사용하여 구성 파일 및 구성 맵을 생성하여 OpenShift Container Platform 클러스터 외부의 대상으로 로그를 보낼 수 있습니다. OpenShift Container Platform에서 로그 데이터를 수신하도록 외부 로그 집계기를 구성해야 합니다.



중요

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

Fluentd **forward** 프로토콜을 사용하여 로그를 보내려면 외부 로그 수집기를 가리키는 **secure-forward.conf**라는 구성 파일을 생성합니다. 그런 다음 해당 파일을 사용하여 OpenShift Container Platform에서 로그를 전달할 때 사용하는 **openshift-logging** 프로젝트에서 **secure-forward**라는 구성 맵을 생성합니다.

사전 요구 사항

- 지정된 프로토콜 또는 형식을 사용하여 로깅 데이터를 수신하도록 구성된 로깅 서버가 있어야 합니다.

샘플 Fluentd 구성 파일

```
<store>
  @type forward
  <security>
    self_hostname ${hostname}
    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 "#{ENV['BUFFER_QUEUE_FULL_ACTION'] || 'throw_exception'}"
  </buffer>
  <server>
    host fluent-receiver.example.com
    port 24224
  </server>
</store>
```

프로세스

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 ${hostname}
    shared_key <key> 1
  </security>
  transport tls 2
  tls_verify_hostname <value> 3
  tls_cert_path <path_to_file> 4
  <buffer> 5
    @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
  </buffer>
  <server>
    name 6
    host 7
    hostlabel 8
    port 9
  </server>
  <server> 10
    name
    host
  </server>
```

1. 노드 간 공유 키를 입력합니다.
2. TLS 유효성 검증을 사용하려면 **tls**를 지정합니다.
3. 서버 인증서 호스트 이름을 확인하려면 **true**로 설정합니다. 서버 인증서 호스트 이름을 무시하려면 **false**로 설정합니다.
4. 사설 CA 인증서 파일의 경로를 **/etc/ocp-forward/ca_cert.pem**으로 지정합니다.
5. 필요에 따라 Fluentd 버퍼 매개변수를 지정합니다.
6. 선택적으로 이 서버의 이름을 입력합니다.
7. 서버의 호스트 이름 또는 IP를 지정합니다.
8. 서버의 호스트 레이블을 지정합니다.
9. 서버의 포트를 지정합니다.
10. 선택적으로 추가 서버를 추가합니다. 둘 이상의 서버를 지정하면 **forward**는 이러한 서버 노드를 라운드 로빈 순서로 사용합니다.

상호 TLS(mTLS) 인증을 사용하려면 Fluentd 설명서에서 클라이언트 인증서, 키 매개변수 및 기타 설정에 대한 정보를 참조하십시오.

2. 구성 파일에서 **openshift-logging** 프로젝트에 **secure-forward**라는 구성 맵을 생성합니다.

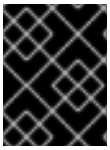
```
$ oc create configmap secure-forward --from-file=secure-forward.conf -n openshift-logging
```

Cluster Logging Operator는 Fluentd Pod를 재배포합니다. Pod가 재배포되지 않으면 Fluentd Pod를 삭제하여 강제로 재배포할 수 있습니다.

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

6.1.7. 레거시 syslog 방법을 사용하여 로그 전달

syslog RFC3164 프로토콜을 사용하여 구성 파일 및 구성 맵을 생성하여 OpenShift Container Platform 클러스터 외부의 대상으로 로그를 보낼 수 있습니다. OpenShift Container Platform에서 로그를 수신하도록 syslog 서버와 같은 외부 로그 수집기를 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 버퍼 없는 구현은 데이터를 버퍼링하지 않고 결과를 즉시 씁니다.
- **out_syslog_buffered**: TCP 및 버퍼를 통해 데이터를 청크로 통신하는 버퍼링된 구현입니다.

syslog 프로토콜을 사용하여 로그를 보내려면 로그를 전달하는 데 필요한 정보와 함께 **syslog.conf**라는 구성 파일을 생성합니다. 그런 다음 해당 파일을 사용하여 OpenShift Container Platform에서 로그를 전달할 때 사용하는 **openshift-logging** 프로젝트에 **syslog**라는 구성 맵을 생성합니다.

사전 요구 사항

- 지정된 프로토콜 또는 형식을 사용하여 로깅 데이터를 수신하도록 구성된 로깅 서버가 있어야 합니다.

샘플 syslog 구성 파일

```
<store>
@type syslog_buffered
remote_syslog rsyslogserver.example.com
port 514
hostname ${hostname}
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 syslog RFC3164를 참조하십시오.

- 기능: syslog 기능. 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 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 메시지의 페이로드로 사용할 레코드 필드입니다.
- rfc: syslog를 사용하여 로그를 보내는 데 사용할 RFC입니다.
- 심각도: 발신 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.

절차

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

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

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

1. 사용할 프로토콜을 **syslog** 또는 **syslog_buffered**로 지정합니다.
2. syslog 서버의 FQDN 또는 IP 주소를 지정합니다.
3. syslog 서버의 포트를 지정합니다.

4 선택 사항: 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.

5 기존 syslog 방법을 사용하여 **rfc** 값으로 **3164**를 지정해야 합니다.

2. 구성 파일에서 **openshift-logging** 프로젝트에 **syslog**라는 구성 맵을 생성합니다.

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

Cluster Logging Operator는 Fluentd Pod를 재배포합니다. Pod가 재배포되지 않으면 Fluentd Pod를 삭제하여 강제로 재배포할 수 있습니다.

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

7장. 쿠버네티스 이벤트 수집 및 저장

OpenShift Container Platform 이벤트 라우터는 Kubernetes 이벤트를 감시하고 클러스터 로깅에 따른 수집을 위해 이러한 이벤트를 기록하는 Pod입니다. 이벤트 라우터를 수동으로 배포해야 합니다.

이벤트 라우터는 모든 프로젝트에서 이벤트를 수집하여 **STDOUT**에 씁니다. 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.

7.1. 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 생성합니다. 템플릿은 또한 이벤트 라우터 Pod를 구성하고 배포합니다. 변경하지 않고 이 템플릿을 사용하거나 배포 오브젝트 CPU 및 메모리 요청을 변경할 수 있습니다.

사전 요구 사항

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

프로세스

1. 이벤트 라우터용 템플릿을 생성합니다.

```
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. **openshift-logging** 프로젝트에서 이벤트 라우터용 서비스 계정을 생성합니다.
2. 클러스터의 이벤트를 모니터링할 ClusterRole을 생성합니다.
3. ClusterRole을 서비스 계정에 바인딩하는 ClusterRoleBinding을 생성합니다.
4. **openshift-logging** 프로젝트에서 구성 맵을 생성하여 필요한 **config.json** 파일을 생성합니다.
5. **openshift-logging** 프로젝트에서 배포를 생성하여 이벤트 라우터 Pod를 생성하고 구성합니다.
6. 이벤트 라우터 Pod에 할당할 최소 메모리 양을 지정합니다. 기본값은 **128Mi**입니다.
7. 이벤트 라우터 Pod에 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 -n openshift-logging -f -
```

예를 들면 다음과 같습니다.

```
$ oc process -f eventrouter.yaml | oc apply -n openshift-logging -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 Pod 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.

8장. 클러스터 로깅 배포

OpenShift Container Platform 클러스터를 4.4에서 4.5로 업데이트한 후 OpenShift Elasticsearch Operator 및 Cluster Logging 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.

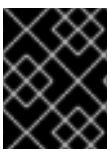
8.1. 클러스터 로깅 배포

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

업데이트하는 경우:

- Cluster Logging Operator를 업데이트하기 전에 OpenShift Elasticsearch Operator를 업데이트해야 합니다.
- OpenShift Elasticsearch Operator와 Cluster Logging Operator를 모두 업데이트해야 합니다. OpenShift Elasticsearch Operator를 업데이트하고 Cluster Logging Operator는 업데이트하지 않으면 Kibana를 사용할 수 없습니다.

  OpenShift Elasticsearch Operator보다 Cluster Logging Operator를 먼저 업데이트하면 Kibana가 업데이트되지 않고 Kibana 사용자 정의 리소스(CR)가 생성되지 않습니다. 이 문제를 해결하려면 Cluster Logging Operator Pod를 삭제하십시오. Cluster Logging Operator Pod가 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 확인합니다.
  - 모든 Pod가 **ready** 상태입니다.
  - Elasticsearch 클러스터는 정상입니다.
- Elasticsearch 및 Kibana 데이터를 백업합니다.

프로세스

1. OpenShift Elasticsearch Operator를 업데이트합니다.

   a. 웹 콘솔에서 **Operator → 설치된 Operator**를 클릭합니다.

   b. **openshift-operators-redhat** 프로젝트를 선택합니다.

   c. **OpenShift Elasticsearch Operator**를 클릭합니다.

   d. **서브스크립션 → 채널**을 클릭합니다.

   e. **서브스크립션 업데이트 채널 변경** 창에서 **4.6**을 선택하고 **저장**을 클릭합니다.

   f. 몇 초 정도 기다린 후 **Operator → 설치된 Operator**를 클릭합니다.
   OpenShift Elasticsearch Operator가 4.6으로 표시됩니다. 예를 들면 다음과 같습니다.

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

   상태 필드가 **성공**으로 표시될 때까지 기다립니다.

2. Cluster Logging Operator 업데이트:

   a. 웹 콘솔에서 **Operator → 설치된 Operator**를 클릭합니다.

   b. **openshift-logging** 프로젝트를 선택합니다.

   c. **Cluster Logging Operator**를 클릭합니다.

   d. **서브스크립션 → 채널**을 클릭합니다.

   e. **서브스크립션 업데이트 채널 변경** 창에서 **4.6**을 선택하고 **저장**을 클릭합니다.

   f. 몇 초 정도 기다린 후 **Operator → 설치된 Operator**를 클릭합니다.
   Cluster Logging Operator가 4.6으로 표시됩니다. 예를 들면 다음과 같습니다.

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

   상태 필드가 **성공**으로 표시될 때까지 기다립니다.

3. 로깅 구성 요소를 확인합니다.

   a. 모든 Elasticsearch Pod가 **ready** 상태인지 확인합니다.

   ```
   $ 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 확인합니다.

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

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

c. Elasticsearch Cron 작업이 생성되었는지 확인합니다.

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

```
$ oc get cronjob
```

```
NAME                     SCHEDULE       SUSPEND   ACTIVE   LAST SCHEDULE   AGE
curator                  30 3,9,15,21 * * * False 0        <none>          20s
elasticsearch-im-app     */15 * * * *   False     0        <none>          56s
elasticsearch-im-audit   */15 * * * *   False     0        <none>          56s
elasticsearch-im-infra   */15 * * * *   False     0        <none>          56s
```

d. 로그 저장소가 4.6으로 업데이트되고 인덱스가 **녹색**인지 확인합니다.

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

출력에 **app-00000x**, **infra-00000x**, **audit-00000x**, **.security** 인덱스가 포함되어 있는지 확인합니다.

예 8.1. 인덱스가 녹색 상태인 샘플 출력

```
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
1SJdCqlZTPWlIAaOUd78yg   1 1          4          0         0            0
green  open   infra-000010
iXwL3bnqTuGEABbUDa6OVw   3 1      248368          0       317          158
green  open   infra-000009
YN9EsULWSNaxWeeNvOs0RA   3 1      258799          0       337          168
green  open   infra-000014
YP0U6R7FQ_GVQVQZ6Yh9Ig   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 infra-000003 goREK1QUKlQPAIVkWVaQ 3 1 226719 0 295 147
green open .security zeT65uOuRTKZMjg_bbUc1g 1 1 5 0 0 0
green open .kibana-377444158_kubeadmin wvMhDwJkR-mRZQO84K0gUQ 3 1 1 0 0 0
green open infra-000006 5H-KBSXGQKiO7hdapDE23g 3 1 226676 0 295 147
green open infra-000001 eH53BQ-bSxSWR5xYZB6lVg 3 1 341800 0 443 220
green open .kibana-6 RVp7TemSSemGJcsSUmuf3A 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 infra-000013 KR9mMFUpQl-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
```

출력에 **fluentd-init** 컨테이너가 포함되어 있는지 확인합니다.

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

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

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

출력에 **ready** 상태가 있는 Kibana pod가 포함되어 있는지 확인합니다.

예 8.2. Kibana Pod가 준비된 샘플 출력

```
[
{
"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-im-app
cronjob.batch/elasticsearch-im-audit
cronjob.batch/elasticsearch-im-infra
```

출력에 **elasticsearch-im-*** 인덱스가 포함되어 있는지 확인합니다.

업데이트 후 작업

Log Forwarding API를 사용하여 로그를 전달하는 경우 OpenShift Elasticsearch Operator 및 Cluster Logging Operator가 4.6으로 완전히 업데이트되면 **LogForwarding 사용자 정의 리소스(CR)를 ClusterLogForwarder CR로** 교체해야 합니다.

8.2. 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 **ClusterLogging** 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 ClusterLogForwarder 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:
     - elasticearch
   - inputSource: logs.app
     name: app-logs
     outputRefs:
      -  elasticearch
   - inputSource: logs.audit
```

```
    name: audit-logs
    outputRefs:
     - fluentd
```

다음 절차는 변경해야 하는 각 매개변수를 보여줍니다.

프로세스

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

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

   샘플 ClusterLogging CR

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  annotations:
    clusterlogging.openshift.io/logforwardingtechpreview: enabled (1)
  name: "instance"
  namespace: "openshift-logging"
....
```

   (1) **logforwardingtechpreview** 주석을 제거합니다.

2. **ClusterLogForwarder** CR을 내보내 **ClusterLogForwarder** 인스턴스의 YAML 파일을 생성합니다.

```
$ 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 (1)
kind: ClusterLogForwarder (2)
metadata:
  name: instance
  namespace: openshift-logging
....
spec: (3)
  outputs:
  - url: http://remote.elasticsearch.com:9200 (4)
    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 유형을 **kind에서 변경합니다. "LogForwarding" ~ kind: "ClusterLogForwarder"**.

3 **disableDefaultForwarding: true** 매개변수를 제거합니다.

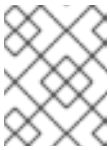
4 출력 매개변수를 **spec.outputs.endpoint**에서 **spec.outputs.url**로 변경합니다. 접두사가 없는 경우 **https://**, **tcp://** 등과 같은 접두사를 URL에 추가합니다.

5 Fluentd 출력의 경우 **유형**을 **forward**에서 **fluentdForward**로 변경합니다.

6 파이프라인 변경:

- **spec.pipelines.inputSource**를 **spec.pipelines.inputRefs**로 변경
- **logs.infra**를 **인프라**로 변경
- **logs.app**을 **애플리케이션**으로 변경
- **logs.audit**를 **감사**로 변경

7 선택 사항: **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
```

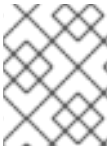
Log Forwarding API의 새로운 기능에 대한 자세한 내용은 타사 시스템에 로그 전달을 참조하십시오.

9장. 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.

9.1. ELASTISEARCH 및 OPENSHIFT LOGGING 대시보드에 액세스

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

프로세스

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/Elastisearch 노드 대시보드 정보 를 참조하십시오.

9.2. OPENSHIFT 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 9.1. OpenShift 로깅 차트

지표	설명
Elastic 클러스터 상태	현재 Elasticsearch 상태: ● 온라인 - Elasticsearch 인스턴스가 온라인 상태임을 나타냅니다. ● 오프라인 - Elasticsearch 인스턴스가 오프라인 상태임을 나타냅니다.
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 대기 시간 합계	● 쿼리 대기 시간: 각 Elasticsearch 검색 쿼리를 실행하는 데 걸리는 평균 시간입니다. ● 가져오기 대기 시간: 각 Elasticsearch 검색 쿼리에서 데이터를 가져오는 데 걸리는 평균 시간입니다. 가져오기 대기 시간은 일반적으로 쿼리 대기 시간보다 더 짧습니다. 가져오기 대기 시간이 지속적으로 증가하는 경우 느린 디스크, 데이터 보강 또는 결과가 너무 많은 대규모 요청을 나타낼 수 있습니다.
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	청크에 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.

9.3. 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 9.2. Elasticsearch 상태 필드

지표	설명

지표	설명
클러스터 상태	Elasticsearch 녹색, 노란색 및 빨간색 상태를 사용하여 선택한 기간 동안의 클러스터 상태: ● 0 - Elasticsearch 인스턴스가 녹색 상태임을 나타냅니다. 이는 모든 shard가 할당되었음을 의미합니다. ● 1 - Elasticsearch 인스턴스가 노란색 상태임을 나타냅니다. 이는 하나 이상의 shard에 대한 복제본 shard가 할당되지 않았음을 의미합니다. ● 2 - Elasticsearch 인스턴스가 빨간색 상태임을 나타냅니다. 이는 하나 이상의 기본 shard와 해당 복제본이 할당되지 않았음을 의미합니다.
클러스터 노드	클러스터의 총 Elasticsearch 노드 수입니다.
클러스터 데이터 노드	클러스터에 있는 Elasticsearch 데이터 노드의 수입니다.
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 수명 주기 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 9.3. 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 재배치 shard	Elasticsearch가 새 노드로 재배치하는 shard 수입니다. Elasticsearch는 노드의 메모리 사용량이 많거나 클러스터에 새 노드를 추가한 경우 등 여러 가지 이유로 노드를 재배치합니다.
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 9.4. 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 메타데이터에 사용되는 총 디스크 공간입니다.
문서 색인 비율	선택한 Elasticsearch 노드에서 문서가 인덱싱되는 비율입니다.

지표	설명
인덱싱 대기 시간	선택한 Elasticsearch 노드에서 문서를 인덱싱하는 데 걸린 시간입니다. 인덱싱 대기 시간은 JVM 힙 메모리 및 전체 로드와 같은 여러 요인의 영향을 받을 수 있습니다. 대기 시간 증가는 인스턴스의 리소스 용량이 부족함을 나타냅니다.
검색률	선택한 Elasticsearch 노드에서 실행되는 검색 요청 수입니다.
검색 대기 시간	선택한 Elasticsearch 노드에서 검색 요청을 완료하는 데 걸린 시간입니다. 검색 대기 시간은 여러 요인의 영향을 받을 수 있습니다. 대기 시간 증가는 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 9.5. 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 9.6. 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 수입니다.
쿼리 캐시 적중	선택한 Elasticsearch 노드의 쿼리 캐시 적중 수입니다.
쿼리 캐시 누락	선택한 Elasticsearch 노드의 쿼리 캐시 누락 수입니다.

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 9.7. 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 9.8. 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 수입니다.

지표	설명
GC 시간	JVM이 선택한 Elasticsearch 노드에서 가비지 수집 작업을 실행하는 데 소비한 시간(오래된 가비지 및 새 가비지 수집 기준)입니다.

10장. 클러스터 로깅 문제 해결

10.1. 클러스터 로깅 상태 보기

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

10.1.1. Cluster Logging Operator의 상태 보기

Cluster Logging Operator의 상태를 볼 수 있습니다.

사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

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: 1
  collection:
    logs:
      fluentdStatus:
        daemonSet: fluentd 2
        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: 3
      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: 4
      kibanaStatus:
      - deployment: kibana
        pods:
          failed: []
          notReady: []
          ready:
          - kibana-7fb4fd4cc9-f2nls
        replicaSets:
        - kibana-7fb4fd4cc9
        replicas: 1
```

1. 출력에서 클러스터 상태 필드가 **상태** 스탠자에 나타납니다.
2. Fluentd Pod에 대한 정보.
3. Elasticsearch 클러스터 건강, **녹색**, **노란색** 또는 **빨간색**을 포함한 Elasticsearch Pod에 대한 정보입니다.
4. Kibana Pod에 대한 정보.

10.1.1.1. 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 않음을 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 높은 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-mkkdys93-1-75dd69dccd-f7f49
        elasticsearch-cdm-mkkdys93-2-67c64f5f4c-n58vl
      Ready:
    Data:
      Failed:
      Not Ready:
        elasticsearch-cdm-mkkdys93-1-75dd69dccd-f7f49
        elasticsearch-cdm-mkkdys93-2-67c64f5f4c-n58vl
      Ready:
    Master:
      Failed:
      Not Ready:
        elasticsearch-cdm-mkkdys93-1-75dd69dccd-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 Pod를 예약할 수 없음을 나타냅니다.

출력 예

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

10.1.2. 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.

프로세스

1. **openshift-logging** 프로젝트로 변경합니다.

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

2. 클러스터 로깅 환경의 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----
```

3. 클러스터 로깅 복제본 세트 상태 보기:

   a. 복제본 세트의 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. 복제본 세트의 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----
```

10.2. 로그 저장소의 상태 보기

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

10.2.1. 로그 저장소의 상태 보기

로그 저장소의 상태를 볼 수 있습니다.

사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

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 Pod를 배치할 수 없는 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.
   - Pod 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 Pod.

10.2.1.1. 상태 메시지 예

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

이 상태 메시지는 노드가 구성된 낮은 워터마크를 초과했으며 이 노드에 shard가 할당되지 않음을 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 높은 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 컨트롤 플레인 노드 (마스터 노드라고도 함)가 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
```

10.2.2. 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 Pod의 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                                 iDjscH7aSUGhIdq0LheLBQ   1
1         5          0         0           0
green  open   .kibana_-377444158_kubeadmin
yBywZ9GfSrKebz5gWBZbjw   3   1         1          0         0           0
green  open   infra-000001                                z6Dpe__ORgiopEpW6Yl44A
3   1    871000          0       874          436
green  open   app-000001                                  hIrazQCeSISewG3c2VIvsQ
3   1      2453          0         3           1
green  open   .kibana_1                                 JCitcBMSQxKOvIq6iQW6wg
1   1         0          0         0           0
green  open   .kibana_-1595131456_user1                        gIYFIEGRRe-
ka0W3okS-mQ   3   1         1          0         0           0
```

로그 저장소 Pod

로그 저장소를 호스팅하는 Pod의 상태를 볼 수 있습니다.

1. Pod 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. Pod 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 스토리지 Pod 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 복제본 세트

로그 저장소 복제본 세트의 상태를 볼 수 있습니다.

1. 복제본 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. 복제본 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>
```

10.3. 클러스터 로깅 경고 이해

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

10.3.1. 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.

프로세스

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 관리를 참조하십시오.

10.3.2. 로깅 수집기 경고 정보

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

표 10.1. Fluentd Prometheus 경고

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

10.3.3. 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 않았습니다.	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. 노드에 디스크 공간을 더 추가하거나 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	Elasticsearch의 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

10.4. 로그 CURATOR 문제 해결

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

10.4.1. 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 작업 실행을 기다리지 않고 로그 레벨을 늘리고 새 작업을 트리거할 수 있습니다.

사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

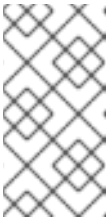
Curator 디버그 로그를 활성화하고 다음 Curator 반복을 수동으로 트리거하려면 다음을 수행합니다.

1. Curator의 디버그 로그를 활성화합니다.

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

로그 수준을 지정합니다.

- **CRITICAL**. Curatior가 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 래퍼 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.

2. 다음 Curator 반복을 트리거합니다.

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

3. 다음 명령을 사용하여 Cron 작업을 제어합니다.
   - Cron 작업 일시 중단:

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

   - Cron 작업 다시 시작:

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

   - Cron 작업 일정 변경:

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

1 **schedule** 옵션은 cron 형식의 일정을 승인합니다.

10.5. RED HAT 지원을 위한 로깅 데이터 수집

지원 사례를 여는 경우 클러스터에 대한 디버깅 정보를 Red Hat 지원에 제공하면 도움이 됩니다.

must-gather 툴을 사용하면 프로젝트 수준 리소스, 클러스터 수준 리소스 및 각 클러스터 로깅 구성 요소에 대한 진단 정보를 수집할 수 있습니다.

즉각 지원을 받을 수 있도록 OpenShift Container Platform 및 클러스터 로깅 둘 다에 대한 진단 정보를 제공하십시오.

참고

hack/logging-dump.sh 스크립트를 사용하지 마십시오. 이 스크립트는 더 이상 지원되지 않으며 데이터를 수집하지 않습니다.

10.5.1. must-gather 툴 정보

oc adm must-gather CLI 명령은 문제를 디버깅하는 데 필요할 가능성이 높은 클러스터에서 정보를 수집합니다.

클러스터 로깅 환경의 경우 **must-gather**는 다음 정보를 수집합니다.

- 프로젝트 수준의 Pod, 구성 맵, 서비스 계정, 역할, 역할 바인딩, 이벤트를 포함한 프로젝트 수준 리소스
- 클러스터 수준의 노드, 역할, 역할 바인딩을 포함한 클러스터 수준 리소스
- 로그 수집기, 로그 저장소, 큐레이터, 로그 시각화 프로그램의 상태를 포함하여 **openshift-logging** 및 **openshift-operators-redhat** 네임스페이스의 클러스터 로깅 리소스

oc adm must-gather를 실행하면 클러스터에 새 Pod가 생성됩니다. 해당 Pod에 대한 데이터가 수집되어 **must-gather.local**로 시작하는 새 디렉터리에 저장됩니다. 이 디렉터리는 현재 작업 중인 디렉터리에 생성되어 있습니다.

10.5.2. 사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

10.5.3. 클러스터 로깅 데이터 수집

oc adm must-gather CLI 명령을 사용하여 클러스터 로깅 환경에 대한 정보를 수집할 수 있습니다.

프로세스

must-gather로 클러스터 로깅 정보를 수집하려면 다음을 수행하십시오.

1. **must-gather** 정보를 저장하려는 디렉터리로 이동합니다.
2. 클러스터 로깅 이미지에 대해 **oc adm must-gather** 명령을 실행합니다.

```
$ oc adm must-gather --image=$(oc -n openshift-logging get deployment.apps/cluster-logging-operator -o jsonpath='{.spec.template.spec.containers[?(@.name == "cluster-logging-operator")].image}')
```

must-gather 툴에서 현재 디렉터리 내에 **must-gather.local**로 시작하는 새 디렉터리를 만듭니다. 예: **must-gather.local.4157245944708210408**.

3. 방금 생성한 **must-gather** 디렉터리에서 압축 파일을 만듭니다. 예를 들어 Linux 운영 체제를 사용하는 컴퓨터에서 다음 명령을 실행합니다.

```
$ tar -cvaf must-gather.tar.gz must-gather.local.4157245944708210408
```

4. Red Hat Customer Portal 에서 해당 지원 사례에 압축 파일을 첨부합니다.

11장. 클러스터 로깅 삭제

OpenShift Container Platform 클러스터에서 클러스터 로깅을 제거할 수 있습니다.

11.1. OPENSHIFT CONTAINER PLATFORM에서 클러스터 로깅 설치 삭제

ClusterLogging 사용자 정의 리소스(CR)를 삭제하여 로그 집계를 중지할 수 있습니다. CR을 삭제한 후에도 다른 클러스터 로깅 구성 요소는 남아 있으며 선택적으로 제거할 수 있습니다.

ClusterLogging CR을 삭제해도 PVC(영구 볼륨 클레임)가 제거되지 않습니다. 나머지 PVC, 영구 볼륨(PV) 및 관련 데이터를 보존하거나 삭제하려면 추가 작업을 수행해야 합니다.

사전 요구 사항

- 클러스터 로깅 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

클러스터 로깅을 제거하려면 다음을 수행합니다.

1. OpenShift Container Platform 웹 콘솔을 사용하여 **ClusterLogging** CR을 제거합니다.
   a. **관리 → 사용자 정의 리소스 정의** 페이지로 전환합니다.
   b. **사용자 정의 리소스 정의** 페이지에서 **ClusterLogging**을 클릭합니다.
   c. **사용자 정의 리소스 정의 세부 정보**페이지에서 **인스턴스**를 클릭합니다.
   d. 인스턴스 옆에 있는 옵션 메뉴  를 클릭하고 **ClusterLogging 삭제**를 선택합니다.
2. 선택 사항: CRD(사용자 정의 리소스 정의)를 삭제합니다.
   a. **관리 → 사용자 정의 리소스 정의** 페이지로 전환합니다.
   b. **ClusterLogForwarder** 옆에 있는 옵션 메뉴  를 클릭하고 **사용자 정의 리소스 정의 삭제**를 선택합니다.
   c. **ClusterLogging** 옆에 있는 옵션 메뉴  를 클릭하고 **사용자 정의 리소스 정의 삭제**를 선택합니다.
   d. **Elasticsearch** 옆에 있는 옵션 메뉴  를 클릭하고 **사용자 정의 리소스 정의 삭제**를 선택합니다.
3. 선택 사항: Cluster Logging Operator 및 OpenShift Elasticsearch Operator를 제거합니다.
   a. **Operator → 설치된 Operator** 페이지로 전환합니다.

b. Cluster Logging Operator 옆에 있는 옵션 메뉴 를 클릭하고 **Operator 설치 제거**를 선택합니다.

c. OpenShift Elasticsearch Operator 옆에 있는 옵션 메뉴 를 클릭하고 **Operator 설치 제거**를 선택합니다.

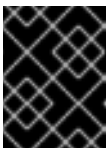
4. 선택 사항: 클러스터 로깅 및 Elasticsearch 프로젝트를 제거합니다.

a. **홈 → 프로젝트** 페이지로 전환합니다.

b. **openshift-logging** 프로젝트 옆에 있는 옵션 메뉴 를 클릭하고 **프로젝트 삭제**를 선택합니다.

c. 대화 상자에서 **openshift-logging**을 입력하여 삭제를 확인하고 **삭제**를 클릭합니다.

d. **openshift-operators-redhat** 프로젝트 옆에 있는 옵션 메뉴 를 클릭하고 **프로젝트 삭제**를 선택합니다.



중요

이 네임스페이스에 다른 글로벌 Operator가 설치된 경우 **openshift-operators-redhat** 프로젝트를 삭제하지 마십시오.

e. 대화 상자에서 **openshift-operators-redhat**을 입력하여 삭제를 확인하고 **삭제**를 클릭합니다.

5. 다른 pod에서 재사용할 수 있도록 PVC를 유지하려면 PVC를 회수하는데 필요한 레이블 또는 PVC 이름을 유지합니다.

6. 선택 사항: PVC를 유지하지 않으려면 삭제할 수 있습니다.

주의

PVC를 해제하거나 삭제하면 PV가 삭제되고 데이터 손실이 발생할 수 있습니다.

a. **스토리지 → 영구 볼륨 클레임** 페이지로 전환합니다.

b. 각 PVC 옆에 있는 옵션 메뉴 를 클릭하고 **영구 볼륨 클레임 삭제**를 선택합니다.

c. 스토리지 공간을 복구하려면 PV를 삭제할 수 있습니다.

추가 리소스

- 수동으로 영구 볼륨 회수

12장. 내보낸 필드

로깅 시스템에서 내보낸 필드이며 Elasticsearch 및 Kibana에서 검색할 수 있습니다. 검색할 때 점으로 구분된 전체 필드 이름을 사용합니다. 예를 들어 Elasticsearch **/_search URL**의 경우 Kubernetes Pod 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**

12.1. 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 **default**의 실제 매핑을 채웁니다.

매개변수	설명
@timestamp	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 **@timestamp**를 찾습니다. 예를 들어 형식은 2015-01-24 14:06:05.071000입니다.
geoip	시스템의 geo-ip입니다.
호스트 이름	**호스트 이름**은 원래 페이로드를 생성하는 엔티티의 FQDN(정규화된 도메인 이름)입니다. 이 필드를 사용하여 이 컨텍스트를 도출하려고 시도합니다. 이를 생성하는 엔티티는 컨텍스트를 알고 있는 경우가 있습니다. 다른 경우에는 해당 엔티티 자체에 제한된 네임스페이스가 있으며 수집기 또는 노멀라이저에서 이를 알고 있습니다.
ipaddr4	소스 서버의 IP 주소 V4. 배열일 수 있습니다.
ipaddr6	사용 가능한 경우 소스 서버의 IP 주소 V6입니다.

매개변수	설명
level	Python의 로깅 모듈인 rsyslog(severitytext 속성)에서 제공하는 로깅 수준입니다. 가능한 값은 **misc/sys/syslog.h**와 **trace** 및 **unknown**에 나와 있습니다. 예를 들면 "alert crit debug emerg err info notice trace unknown warning"입니다. **trace**는 **syslog.h** 목록에 없지만 많은 애플리케이션에서 사용합니다. . 로깅 시스템이 이해하지 못하는 값을 얻을 때만 **unknown** 값을 사용해야 하며 이 값은 최고 수준임을 유의하십시오. **debug**보다 높거나 더 자세한 **trace**를 고려하십시오. **error**는 더 이상 사용되지 않으며**err**을 사용합니다. **panic**을 **emerg**로 변환합니다. **warn**을 **warning**으로 변환합니다. **syslog/journal PRIORITY**의 숫자 값은 일반적으로 misc/sys/syslog.h에 나열된 우선 순위 값을 사용하여 매핑할 수 있습니다. 다른 로깅 시스템의 로그 수준과 우선 순위는 가장 가까운 일치 항목에 매핑해야 합니다. 예제는 python 로깅을 참조하십시오.
message	일반적인 로그 항목 메시지 또는 페이로드입니다. 수집기 또는 노멀라이저에 의해 풀링된 UTF-8 인코딩의 메타데이터를 제거할 수 있습니다.
pid	사용 가능한 경우 이는 로깅 엔티티의 프로세스 ID입니다.
service	사용 가능한 경우 로깅 엔티티와 연관된 서비스의 이름입니다. 예를 들어 **syslog APP-NAME** 속성은 서비스 필드에 매핑됩니다.
tags	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.
file	파일 경로에 대한 수집기 **TODO** 분석기에 로컬인 로그 항목을 포함하는 파일의 선택적 경로입니다.
offset	오프셋 값은 단일 로그 파일의 컨텍스트에서 값이 엄격하게 단조롭게 증가하는 한 파일에서 로그 라인의 시작까지의 바이트 수(0 또는 1 기반) 또는 로그 라인 번호(0 또는 1 기반)를 나타낼 수 있습니다. 새 버전의 로그 파일(회전)을 나타내는 값을 줄바꿈할 수 있습니다.
namespace_name	이 레코드를 이름을 공유하는 **네임스페이스** 와 연결합니다. 이 값은 저장되지 않지만 레코드를 액세스 제어 및 시각화를 위한 적절한 **네임스페이스** 와 연결하는 데 사용됩니다. 일반적으로 이 값은 태그에 제공되지만 프로토콜이 태그 전송을 지원하지 않으면 이 필드를 사용할 수 있습니다. 이 필드가 있으면 태그 또는 **kubernetes.namespace_name**에 지정된 **네임스페이스**를 대체합니다.
namespace_uuid	이는 **namespace_name**과 연관된 **uuid**입니다. 이 값은 저장되지 않지만 레코드를 액세스 제어 및 시각화를 위한 적절한 네임스페이스와 연결하는 데 사용됩니다. 이 필드가 있으면 **kubernetes.namespace_uuid**에 지정된 **uuid**를 대체합니다. 또한 이 로그 레코드에 대해 쿠버네티스 메타데이터 조회를 건너뜁니다.

collectd 필드
다음 필드는 네임스페이스 지표 메타데이터를 나타냅니다.

매개변수	설명
collectd.interval	유형: 플로트 **collectd** 간격.
collectd.plugin	유형: 문자열 **collectd** 플러그인.
collectd.plugin_instance	유형: 문자열 **collectd** plugin_instance.
collectd.type_instance	유형: 문자열 **collectd type_instance**.
collectd.type	유형: 문자열 **collectd** 유형.
collectd.dstypes	유형: 문자열 **collectd** dstypes.

collectd.processes 필드
다음 필드는 **collectd** 프로세스 플러그인에 해당합니다.

매개변수	설명
collectd.processes.ps_state	유형: 정수 프로세스 플러그인의 **collectd ps_state** 유형.

collectd.processes.ps_disk_ops 필드
프로세스 플러그인의 **collectd ps_disk_ops** 유형.

매개변수	설명
collectd.processes.ps_disk_ops.read	유형: 플로트 **TODO**
collectd.processes.ps_disk_ops.write	유형: 플로트 **TODO**

매개변수	설명
collectd.processes.ps_vm	유형: 정수 프로세스 플러그인의 **collectd ps_vm** 유형.
collectd.processes.ps_rss	유형: 정수 프로세스 플러그인의 **collectd ps_rss** 유형.
collectd.processes.ps_data	유형: 정수 프로세스 플러그인의 **collectd ps_data** 유형.
collectd.processes.ps_code	유형: 정수 프로세스 플러그인의 **collectd ps_code** 유형.
collectd.processes.ps_stacksize	유형: 정수 프로세스 플러그인의 **collectd ps_stacksize** 유형.

collectd.processes.ps_cputime 필드
프로세스 플러그인의 **collectd ps_cputime** 유형.

매개변수	설명
collectd.processes.ps_cputime.user	유형: 플로트 **TODO**
collectd.processes.ps_cputime.syst	유형: 플로트 **TODO**

collectd.processes.ps_count 필드
프로세스 플러그인의 **collectd ps_count** 유형.

매개변수	설명
collectd.processes.ps_count.processes	유형: 정수 **TODO**
collectd.processes.ps_count.threads	유형: 정수 **TODO**

collectd.processes.ps_pagefaults 필드

프로세스 플러그인의 **collectd ps_pagefaults** 유형.

매개변수	설명
collectd.processes.ps_pagefaults.majflt	유형: 플로트 **TODO**
collectd.processes.ps_pagefaults.minflt	유형: 플로트 **TODO**

collectd.processes.ps_disk_octets 필드
프로세스 플러그인의 **collectd ps_disk_octets** 유형.

매개변수	설명
collectd.processes.ps_disk_octets.read	유형: 플로트 **TODO**
collectd.processes.ps_disk_octets.write	유형: 플로트 **TODO**
collectd.processes.fork_rate	유형: 플로트 프로세스 플러그인의 **collectd fork_rate** 유형.

collectd.disk 필드
collectd 디스크 플러그인에 해당합니다.

collectd.disk.disk_merged 필드
디스크 플러그인의 **collectd disk_merged** 유형.

매개변수	설명
collectd.disk.disk_merged.read	유형: 플로트 **TODO**
collectd.disk.disk_merged.write	유형: 플로트 **TODO**

collectd.disk.disk_octets 필드
디스크 플러그인의 **collectd disk_octets** 유형.

매개변수	설명
collectd.disk.disk_octets.read	유형: 플로트 **TODO**
collectd.disk.disk_octets.write	유형: 플로트 **TODO**

collectd.disk.disk_time 필드
디스크 플러그인의 **collectd disk_time** 유형.

매개변수	설명
collectd.disk.disk_time.read	유형: 플로트 **TODO**
collectd.disk.disk_time.write	유형: 플로트 **TODO**

collectd.disk.disk_ops 필드
디스크 플러그인의 **collectd disk_ops** 유형.

매개변수	설명
collectd.disk.disk_ops.read	유형: 플로트 **TODO**
collectd.disk.disk_ops.write	유형: 플로트 **TODO**
collectd.disk.pending_operations	유형: 정수 디스크 플러그인의 **collectd pending_operations** 유형.

collectd.disk.disk_io_time 필드
디스크 플러그인의 **collectd disk_io_time** 유형.

매개변수	설명
collectd.disk.disk_io_time.io_time	유형: 플로트 **TODO**

매개변수	설명
collectd.disk.disk_io_time .weighted_io_time	유형: 플로트 **TODO**

collectd.interface 필드
collectd 인터페이스 플러그인에 해당합니다.

collectd.interface.if_octets 필드
인터페이스 플러그인의 **collectd if_octets** 유형.

매개변수	설명
collectd.interface.if_octets.rx	유형: 플로트 **TODO**
collectd.interface.if_octets.tx	유형: 플로트 **TODO**

collectd.interface.if_packets 필드
인터페이스 플러그인의 **collectd if_packets** 유형.

매개변수	설명
collectd.interface.if_packets.rx	유형: 플로트 **TODO**
collectd.interface.if_packets.tx	유형: 플로트 **TODO**

collectd.interface.if_errors 필드
인터페이스 플러그인의 **collectd if_errors** 유형.

매개변수	설명
collectd.interface.if_errors.rx	유형: 플로트 **TODO**
collectd.interface.if_errors.tx	유형: 플로트 **TODO**

collectd.interface.if_dropped 필드
인터페이스 플러그인의 **collectd if_dropped** 유형.

매개변수	설명
collectd.interface.if_dropped.rx	유형: 플로트 **TODO**
collectd.interface.if_dropped.tx	유형: 플로트 **TODO**

collectd.virt 필드
collectd 가상화 플러그인에 해당합니다.

collectd.virt.if_octets 필드
가상화 플러그인의 **collectd if_octets** 유형.

매개변수	설명
collectd.virt.if_octets.rx	유형: 플로트 **TODO**
collectd.virt.if_octets.tx	유형: 플로트 **TODO**

collectd.virt.if_packets 필드
가상화 플러그인의 **collectd if_packets** 유형.

매개변수	설명
collectd.virt.if_packets.rx	유형: 플로트 **TODO**
collectd.virt.if_packets.tx	유형: 플로트 **TODO**

collectd.virt.if_errors 필드
가상화 플러그인의 **collectd if_errors** 유형.

매개변수	설명

매개변수	설명
collectd.virt.if_errors.rx	유형: 플로트 **TODO**
collectd.virt.if_errors.tx	유형: 플로트 **TODO**

collectd.virt.if_dropped 필드
가상화 플러그인의 **collectd if_dropped** 유형.

매개변수	설명
collectd.virt.if_dropped.rx	유형: 플로트 **TODO**
collectd.virt.if_dropped.tx	유형: 플로트 **TODO**

collectd.virt.disk_ops 필드
가상화 플러그인의 **collectd disk_ops** 유형.

매개변수	설명
collectd.virt.disk_ops.read	유형: 플로트 **TODO**
collectd.virt.disk_ops.write	유형: 플로트 **TODO**

collectd.virt.disk_octets 필드
가상화 플러그인의 **collectd disk_octets** 유형.

매개변수	설명
collectd.virt.disk_octets.read	유형: 플로트 **TODO**
collectd.virt.disk_octets.write	유형: 플로트 **TODO**

매개변수	설명
collectd.virt.memory	유형: 플로트 가상화 플러그인의 **collectd** 메모리 유형.
collectd.virt.virt_vcpu	유형: 플로트 가상화 플러그인의 **collectd virt_vcpu** 유형.
collectd.virt.virt_cpu_total	유형: 플로트 가상화 플러그인의 **collectd virt_cpu_total** 유형.

collectd.CPU 필드
collectd CPU 플러그인에 해당합니다.

매개변수	설명
collectd.CPU.percent	유형: 플로트 플러그인 CPU의 **collectd** 유형 백분율.

collectd.df 필드
collectd df 플러그인에 해당합니다.

매개변수	설명
collectd.df.df_complex	유형: 플로트 플러그인 **df**의 **collectd** 유형 **df_complex**.
collectd.df.percent_bytes	유형: 플로트 플러그인 **df**의 **collectd** 유형 **percent_bytes**.

collectd.entropy 필드
collectd 엔트로피 플러그인에 해당합니다.

매개변수	설명
collectd.entropy.entropy	유형: 정수 엔트로피 플러그인의 **collectd** 엔트로피 유형.

collectd.memory 필드
collectd 메모리 플러그인에 해당합니다.

매개변수	설명
collectd.memory.memory	유형: 플로트 메모리 플러그인의 **collectd** 메모리 유형.
collectd.memory.percent	유형: 플로트 메모리 플러그인의 **collectd** 백분율 유형.

collectd.swap 필드
collectd 스왑 플러그인에 해당합니다.

매개변수	설명
collectd.swap.swap	유형: 정수 스왑 플러그인의 **collectd** 스왑 유형.
collectd.swap.swap_io	유형: 정수 스왑 플러그인의 **collectd swap_io** 유형.

collectd.load 필드
collectd 로드 플러그인에 해당합니다.

collectd.load.load 필드
로드 플러그인의 **collectd** 로드 유형.

매개변수	설명
collectd.load.load.shortterm	유형: 플로트 **TODO**
collectd.load.load.midterm	유형: 플로트 **TODO**
collectd.load.load.longterm	유형: 플로트 **TODO**

collectd.aggregation 필드
collectd 집계 플러그인에 해당합니다.

매개변수	설명
collectd.aggregation.percent	유형: 플로트 **TODO**

collectd.statsd 필드
collectd statsd 플러그인에 해당합니다.

매개변수	설명
collectd.statsd.host_cpu	유형: 정수 **statsd** 플러그인의 **collectd** CPU 유형.
collectd.statsd.host_elapsed_time	유형: 정수 **statsd** 플러그인의 **collectd elapsed_time** 유형.
collectd.statsd.host_memory	유형: 정수 **statsd** 플러그인의 **collectd** 메모리 유형.
collectd.statsd.host_nic_speed	유형: 정수 **statsd** 플러그인의 **collectd nic_speed** 유형.
collectd.statsd.host_nic_rx	유형: 정수 **statsd** 플러그인의 **collectd nic_rx** 유형.
collectd.statsd.host_nic_tx	유형: 정수 **statsd** 플러그인의 **collectd nic_tx** 유형.
collectd.statsd.host_nic_rx_dropped	유형: 정수 **statsd** 플러그인의 **collectd nic_rx_dropped** 유형.
collectd.statsd.host_nic_tx_dropped	유형: 정수 **statsd** 플러그인의 **collectd nic_tx_dropped** 유형.
collectd.statsd.host_nic_rx_errors	유형: 정수 **statsd** 플러그인의 **collectd nic_rx_errors** 유형.
collectd.statsd.host_nic_tx_errors	유형: 정수 **statsd** 플러그인의 **collectd nic_tx_errors** 유형.

매개변수	설명
collectd.statsd.host_storage	유형: 정수 **statsd** 플러그인의 **collectd** 스토리지 유형.
collectd.statsd.host_swap	유형: 정수 **statsd** 플러그인의 **collectd** 스왑 유형.
collectd.statsd.host_vdsm	유형: 정수 **statsd** 플러그인의 **collectd** VDSM 유형.
collectd.statsd.host_vms	유형: 정수 **statsd** 플러그인의 **collectd** VMS 유형.
collectd.statsd.vm_nic_tx_dropped	유형: 정수 **statsd** 플러그인의 **collectd nic_tx_dropped** 유형.
collectd.statsd.vm_nic_rx_bytes	유형: 정수 **statsd** 플러그인의 **collectd nic_rx_bytes** 유형.
collectd.statsd.vm_nic_tx_bytes	유형: 정수 **statsd** 플러그인의 **collectd nic_tx_bytes** 유형.
collectd.statsd.vm_balloon_min	유형: 정수 **statsd** 플러그인의 **collectd balloon_min** 유형.
collectd.statsd.vm_balloon_max	유형: 정수 **statsd** 플러그인의 **collectd balloon_max** 유형.
collectd.statsd.vm_balloon_target	유형: 정수 **statsd** 플러그인의 **collectd balloon_target** 유형.
collectd.statsd.vm_balloon_cur	유형: 정수 **statsd** 플러그인의 **collectd balloon_cur** 유형.
collectd.statsd.vm_cpu_sys	유형: 정수 **statsd** 플러그인의 **collectd cpu_sys** 유형.

매개변수	설명
collectd.statsd.vm_cpu_usage	유형: 정수 **statsd** 플러그인의 **collectd cpu_usage** 유형.
collectd.statsd.vm_disk_read_ops	유형: 정수 **statsd** 플러그인의 **collectd disk_read_ops** 유형.
collectd.statsd.vm_disk_write_ops	유형: 정수 **statsd** 플러그인의 **collectd disk_write_ops** 유형.
collectd.statsd.vm_disk_flush_latency	유형: 정수 **statsd** 플러그인의 **collectd disk_flush_latency** 유형.
collectd.statsd.vm_disk_apparent_size	유형: 정수 **statsd** 플러그인의 **collectd disk_apparent_size** 유형.
collectd.statsd.vm_disk_write_bytes	유형: 정수 **statsd** 플러그인의 **collectd disk_write_bytes** 유형.
collectd.statsd.vm_disk_write_rate	유형: 정수 **statsd** 플러그인의 **collectd disk_write_rate** 유형.
collectd.statsd.vm_disk_true_size	유형: 정수 **statsd** 플러그인의 **collectd disk_true_size** 유형.
collectd.statsd.vm_disk_read_rate	유형: 정수 **statsd** 플러그인의 **collectd disk_read_rate** 유형.
collectd.statsd.vm_disk_write_latency	유형: 정수 **statsd** 플러그인의 **collectd disk_write_latency** 유형.
collectd.statsd.vm_disk_read_latency	유형: 정수 **statsd** 플러그인의 **collectd disk_read_latency** 유형
collectd.statsd.vm_disk_read_bytes	유형: 정수 **statsd** 플러그인의 **collectd disk_read_bytes** 유형.

매개변수	설명
collectd.statsd.vm_nic_rx_dropped	유형: 정수 **statsd** 플러그인의 **collectd nic_rx_dropped** 유형.
collectd.statsd.vm_cpu_user	유형: 정수 **statsd** 플러그인의 **collectd cpu_user** 유형.
collectd.statsd.vm_nic_rx_errors	유형: 정수 **statsd** 플러그인의 **collectd nic_rx_errors** 유형.
collectd.statsd.vm_nic_tx_errors	유형: 정수 **statsd** 플러그인의 **collectd nic_tx_errors** 유형.
collectd.statsd.vm_nic_speed	유형: 정수 **statsd** 플러그인의 **collectd nic_speed** 유형.

collectd.postgresql 필드
collectd postgresql 플러그인에 해당합니다.

매개변수	설명
collectd.postgresql.pg_n_tup_g	유형: 정수 플러그인 postgresql의 **collectd** 유형 **pg_n_tup_g**
collectd.postgresql.pg_n_tup_c	유형: 정수 플러그인 postgresql의 **collectd** 유형 **pg_n_tup_c**
collectd.postgresql.pg_numbackends	유형: 정수 플러그인 postgresql의 **collectd** 유형 **pg_numbackends**
collectd.postgresql.pg_xact	유형: 정수 플러그인 postgresql의 **collectd** 유형 **pg_xact**
collectd.postgresql.pg_db_size	유형: 정수 플러그인 postgresql의 **collectd** 유형 **pg_db_size**
collectd.postgresql.pg_blks	유형: 정수 플러그인 postgresql의 **collectd** 유형 **pg_blks**

12.2. SYSTEMD 내보낸 필드

OpenShift Container Platform 클러스터 로깅에서 보낸 **systemd** 필드로 Elasticsearch와 Kibana에서 검색할 수 있습니다.

systemd 저널에 해당하는 공통 필드를 포함합니다. 애플리케이션은 자체 필드를 저널에 쓸 수 있습니다. **systemd.u** 네임스페이스에서 사용할 수 있습니다. 해당하는 필드는 **RESULT** 및 **UNIT**입니다.

systemd.k 필드
다음 표에는 **systemd** 커널 특정 메타데이터가 포함되어 있습니다.

매개변수	설명
systemd.k.KERNEL_DEVICE	**systemd.k.KERNEL_DEVICE**는 커널 장치 이름입니다.
systemd.k.KERNEL_SUBSYSTEM	**systemd.k.KERNEL_SUBSYSTEM**은 커널 하위 시스템 이름입니다.
systemd.k.UDEV_DEVLINK	**systemd.k.UDEV_DEVLINK**에는 노드를 가리키는 추가 기호 링크 이름이 포함되어 있습니다.
systemd.k.UDEV_DEVNODE	**systemd.k.UDEV_DEVNODE**는 장치의 노드 경로입니다.
systemd.k.UDEV_SYSNAME	**systemd.k.UDEV_SYSNAME**은 커널 장치 이름입니다.

systemd.t 필드
systemd.t 필드는 신뢰할 수 있는 저널 필드, 저널이 암시적으로 추가한 필드이며 클라이언트 코드로 변경할 수 없습니다.

매개변수	설명
systemd.t.AUDIT_LOGINUID	**systemd.t.AUDIT_LOGINUID**는 저널 프로세스의 사용자 ID입니다.
systemd.t.BOOT_ID	**systemd.t.BOOT_ID**는 커널 부팅 ID입니다.
systemd.t.AUDIT_SESSION	**systemd.t.AUDIT_SESSION**은 저널 프로세스의 세션입니다.
systemd.t.CAP_EFFECTIVE	**systemd.t.CAP_EFFECTIVE**는 저널 프로세스의 기능을 나타냅니다.
systemd.t.CMDLINE	**systemd.t.CMDLINE**은 저널 프로세스의 명령줄입니다.
systemd.t.COMM	**systemd.t.COMM**은 저널 입력 프로세스의 이름입니다.

매개변수	설명
systemd.t.EXE	**systemd.t.EXE**는 저널 입력 프로세스의 실행 가능 경로입니다.
systemd.t.GID	**systemd.t.GID**는 저널 입력 프로세스의 그룹 ID입니다.
systemd.t.HOSTNAME	**systemd.t.HOSTNAME**은 호스트 이름입니다.
systemd.t.MACHINE_ID	**systemd.t.MACHINE_ID**는 호스트의 시스템 ID입니다.
systemd.t.PID	**systemd.t.PID**는 저널 입력 프로세스의 프로세스 ID입니다.
systemd.t.SELINUX_CONTEXT	**systemd.t.SELINUX_CONTEXT**는 저널 입력 프로세스의 보안 컨텍스트 또는 레이블입니다.
systemd.t.SOURCE_REALTIME_TIMESTAMP	**systemd.t.SOURCE_REALTIME_TIMESTAMP**는 메시지의 가장 빠르고 가장 안정적인 타임스탬프입니다. 이 타임스탬프는 RFC 3339 NS 형식으로 변환됩니다.
systemd.t.SYSTEMD_CGROUP	**systemd.t.SYSTEMD_CGROUP**는 **systemd** 제어 그룹 경로입니다.
systemd.t.SYSTEMD_OWNER_UID	**systemd.t.SYSTEMD_OWNER_UID**는 세션의 소유자 ID입니다.
systemd.t.SYSTEMD_SESSION	해당하는 경우 **systemd.t.SYSTEMD_SESSION**은 **systemd** 세션 ID입니다.
systemd.t.SYSTEMD_SLICE	**systemd.t.SYSTEMD_SLICE**는 저널 입력 프로세스의 슬라이스 단위입니다.
systemd.t.SYSTEMD_UNIT	**systemd.t.SYSTEMD_UNIT**는 세션의 단위 이름입니다.
systemd.t.SYSTEMD_USER_UNIT	해당하는 경우 **systemd.t.SYSTEMD_USER_UNIT**는 세션의 사용자 단위 이름입니다.
systemd.t.TRANSPORT	**systemd.t.TRANSPORT**는 저널 서비스의 입력 방법입니다. 여기에는 **audit**, **driver**, **syslog**, **journal**, **stdout** 및 **kernel**이 포함됩니다.
systemd.t.UID	**systemd.t.UID**는 저널 입력 프로세스의 사용자 ID입니다.
systemd.t.SYSLOG_FACILITY	**systemd.t.SYSLOG_FACILITY**는 **syslog**에 대한 기능을 포함하는 필드이며 10진수 문자열 형식입니다.
systemd.t.SYSLOG_IDENTIFIER	**systemd.t.systemd.t.SYSLOG_IDENTIFIER**는 **syslog**의 식별자입니다.

매개변수	설명
systemd.t.SYSLOG_PID	**SYSLOG_PID**는 **syslog**의 클라이언트 프로세스 ID입니다.

systemd.u 필드
systemd.u 필드는 클라이언트에서 직접 전달되어 저널에 저장됩니다.

매개변수	설명
systemd.u.CODE_FILE	**systemd.u.CODE_FILE**은 소스의 파일 이름이 포함된 코드 위치입니다.
systemd.u.CODE_FUNCTION	**systemd.u.CODE_FUNCTION**은 소스 기능을 포함하는 코드 위치입니다.
systemd.u.CODE_LINE	**systemd.u.CODE_LINE**은 소스의 라인 번호를 포함하는 코드 위치입니다.
systemd.u.ERRNO	**systemd.u.ERRNO**(있는 경우)는 숫자 값 형식의 하위 수준 오류 번호를 나타내며 10진수 문자열입니다.
systemd.u.MESSAGE_ID	**systemd.u.MESSAGE_ID**는 메시지 유형을 인식하기 위한 메시지 식별자 ID입니다.
systemd.u.RESULT	개인 용도로만 사용하십시오.
systemd.u.UNIT	개인 용도로만 사용하십시오.

12.3. 쿠버네티스 내보낸 필드

다음은 OpenShift Container Platform 클러스터 로깅에서 내보낸 쿠버네티스 필드로 Elasticsearch 및 Kibana에서 검색할 수 있습니다.

쿠버네티스 관련 메타데이터의 네임스페이스입니다. **kubernetes.pod_name**은 Pod의 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 교체됩니다.

매개변수	설명
kubernetes.pod_id	Pod의 쿠버네티스 ID입니다.
kubernetes.namespace_name	쿠버네티스의 네임스페이스 이름입니다.
kubernetes.namespace_id	쿠버네티스의 네임스페이스 ID입니다.

매개변수	설명
kubernetes.host	쿠버네티스 노드 이름입니다.
kubernetes.container_name	쿠버네티스의 컨테이너 이름입니다.
kubernetes.labels.deployment	쿠버네티스 오브젝트와 관련된 배포입니다.
kubernetes.labels.deploymentconfig	쿠버네티스 오브젝트와 관련된 배포 구성입니다.
kubernetes.labels.component	쿠버네티스 오브젝트와 관련된 구성 요소입니다.
kubernetes.labels.provider	쿠버네티스 오브젝트와 관련된 공급자입니다.

kubernetes.annotations 필드
OpenShift 오브젝트와 관련된 주석은 **kubernetes.annotations** 필드입니다.

12.4. 컨테이너 내보낸 필드

OpenShift Container Platform 클러스터 로깅에서 내보낸 Docker 필드로 Elasticsearch 및 Kibana에서 검색할 수 있습니다. Docker 컨테이너 특정 메타데이터의 네임스페이스입니다. docker.container_id는 Docker 컨테이너 ID입니다.

pipeline_metadata.collector 필드
이 섹션에는 수집기 고유의 메타데이터가 포함되어 있습니다.

매개변수	설명
pipeline_metadata.collector.hostname	수집기의 FQDN입니다. 실제 로그 송신기의 FQDN과 다를 수 있습니다.
pipeline_metadata.collector.name	수집기의 이름입니다.
pipeline_metadata.collector.version	수집기의 버전입니다.
pipeline_metadata.collector.ipaddr4	수집기 서버의 IP 주소 v4는 배열일 수 있습니다.
pipeline_metadata.collector.ipaddr6	수집기 서버의 IP 주소 v6은 배열일 수 있습니다.

매개변수	설명
pipeline_metadata.collector.inputname	TCP/UDP 또는 imjournal/imfile과 같이 수집기에서 로그 메시지를 수신한 방법입니다.
pipeline_metadata.collector.received_at	수집기가 메시지를 수신한 시간입니다.
pipeline_metadata.collector.original_raw_message	수집기에서 수집하거나 소스와 최대한 가까운 구문 분석되지 않은 원본 로그 메시지입니다.

pipeline_metadata.normalizer 필드
이 섹션에는 노멀라이저 특정 메타데이터가 포함되어 있습니다.

매개변수	설명
pipeline_metadata.normalizer.hostname	노멀라이저의 FQDN입니다.
pipeline_metadata.normalizer.name	노멀라이저의 이름입니다.
pipeline_metadata.normalizer.version	노멀라이저의 버전입니다.
pipeline_metadata.normalizer.ipaddr4	노멀라이저 서버의 IPv4 주소이며 배열일 수 있습니다.
pipeline_metadata.normalizer.ipaddr6	노멀라이저 서버의 IPv6 주소이며 배열일 수 있습니다.
pipeline_metadata.normalizer.inputname	TCP/UDP 등 노멀라이저가 로그 메시지를 수신한 방법입니다.
pipeline_metadata.normalizer.received_at	노멀라이저가 메시지를 수신한 시간입니다.
pipeline_metadata.normalizer.original_raw_message	노멀라이저가 수신한 구문 분석되지 않은 원본 로그 메시지입니다.
pipeline_metadata.trace	이 필드는 메시지 추적을 기록합니다. 각 수집기 및 노멀라이저는 자체 정보와 메시지가 처리된 날짜 및 시간을 추가합니다.

12.5. OVIRT 내보낸 필드

OpenShift Container Platform 클러스터 로깅에서 내보낸 oVirt 필드로 Elasticsearch 및 Kibana에서 검색할 수 있습니다.

oVirt 메타데이터의 네임스페이스입니다.

매개변수	설명
ovirt.entity	데이터 소스, 호스트, VMS 및 엔진의 유형입니다.
ovirt.host_id	oVirt 호스트 UUID입니다.

ovirt.engine 필드
관리자와 관련된 메타데이터의 네임스페이스입니다. Manager의 FQDN은 **ovirt.engine.fqdn입니다.**

12.6. AUSHAPE 내보낸 필드

OpenShift Container Platform 클러스터 로깅에서 내보낸 Aushape 필드로 Elasticsearch 및 Kibana에서 검색할 수 있습니다.

Aushape로 변환된 감사 이벤트입니다. 자세한 내용은 Aushape를 참조하십시오.

매개변수	설명
aushape.serial	감사 이벤트 일련 번호입니다.
aushape.node	감사 이벤트가 발생한 호스트의 이름입니다.
aushape.error	이벤트를 변환하는 중에 aushape에 발생한 오류입니다.
aushape.trimmed	이벤트 오브젝트를 기준으로 한 JSONPath 표현식의 배열로, 이벤트 크기 제한에 따라 컨텐츠가 제거된 오브젝트 또는 배열을 지정합니다. 빈 문자열은 이벤트가 내용을 제거했음을 의미하고 빈 배열은 지정되지 않은 오브젝트 및 배열에 의해 트리밍이 발생했음을 의미합니다.
aushape.text	원래 감사 이벤트를 나타내는 배열 로그 레코드 문자열입니다.

aushape.data 필드
Aushape와 관련된 감사 이벤트 데이터를 구문 분석했습니다.

매개변수	설명
aushape.data.avc	유형: 중첩
aushape.data.execve	유형: 문자열
aushape.data.netfilter_cfg	유형: 중첩
aushape.data.obj_pid	유형: 중첩
aushape.data.path	유형: 중첩

12.7. TLOG 내보낸 필드

OpenShift Container Platform 클러스터 로깅 시스템에서 내보낸 Tlog 필드로 Elasticsearch 및 Kibana에서 검색할 수 있습니다.

Tlog 터미널 I/O 기록 메시지입니다. 자세한 내용은 Tlog를 참조하십시오.

매개변수	설명
tlog.ver	메시지 형식 버전 번호입니다.
tlog.user	기록된 사용자 이름입니다.
tlog.term	터미널 유형 이름입니다.
tlog.session	기록된 세션의 감사 세션 ID입니다.
tlog.id	세션 내 메시지의 ID입니다.
tlog.pos	세션 내 메시지 위치입니다(밀리초).
tlog.timing	이 메시지의 이벤트를 제시간에 분배합니다.
tlog.in_txt	유효하지 않은 문자가 제거된 입력 텍스트입니다.
tlog.in_bin	유효하지 않은 입력 문자를 바이트로 제거했습니다.
tlog.out_txt	유효하지 않은 문자를 제거한 출력 텍스트입니다.
tlog.out_bin	유효하지 않은 출력 문자를 바이트로 제거했습니다.