



# OpenShift Container Platform 4.7

## 로깅

OpenShift Logging 설치, 사용법, 릴리스 정보



## OpenShift Container Platform 4.7 로깅

---

OpenShift Logging 설치, 사용법, 릴리스 정보

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

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Logging.ent file |.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## 초록

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

## 차례

<b>1장. RED HAT OPENSIFT LOGGING 릴리스 노트</b> .....	<b>7</b>
1.1. 보다 포괄적 수용을 위한 오픈 소스 용어 교체	7
1.2. 지원되는 버전	7
1.2.1. OpenShift Logging 5.1.0	7
1.2.1.1. 새로운 기능 및 개선 사항	7
1.2.1.2. 사용되지 않거나 삭제된 기능	8
1.2.1.2.1. Elasticsearch Curator가 제거됨	8
1.2.1.2.2. 레거시 Fluentd 및 레거시 syslog 방법을 사용하여 로그 전달이 더 이상 사용되지 않음	8
1.2.1.3. 버그 수정	8
1.2.2. OpenShift Logging 5.0.9	9
1.2.2.1. 버그 수정	9
1.2.2.2. CVE	10
1.2.3. OpenShift Logging 5.0.8	10
1.2.3.1. 버그 수정	10
1.2.4. OpenShift Logging 5.0.7	10
1.2.4.1. 버그 수정	10
1.2.4.2. CVE	10
1.2.5. OpenShift Logging 5.0.6	13
1.2.5.1. 버그 수정	13
1.2.5.2. CVE	14
1.2.6. OpenShift Logging 5.0.5	15
1.2.6.1. 보안 수정	16
1.2.7. OpenShift Logging 5.0.4	16
1.2.7.1. 보안 수정	16
1.2.7.2. 버그 수정	16
1.2.8. OpenShift Logging 5.0.3	16
1.2.8.1. 보안 수정	16
1.2.8.2. 버그 수정	18
1.2.9. OpenShift Logging 5.0.2	18
1.2.9.1. 버그 수정	18
1.2.10. OpenShift Logging 5.0.1	19
1.2.10.1. 버그 수정	19
1.2.11. OpenShift Logging 5.0.0	19
1.2.11.1. 새로운 기능 및 개선 사항	19
클러스터 로깅이 Red Hat OpenShift Logging이 됨	19
인덱스당 최대 5개의 기본 샤드	19
OpenShift Elasticsearch Operator 이름 및 수준 업데이트	19
OpenShift Elasticsearch Operator에서 CSV 성공 사례 보고	20
Elasticsearch pod 인증서 권한 경고 감소	20
경고에서 설명 및 문제 해결에 새로운 링크	20
삭제 작업에 대한 새로운 연결 시간 제한	20
롤오버 인덱스 템플릿에 업데이트를 최소화	20
1.2.11.2. 기술 프리뷰 기능	20
1.2.11.3. 사용되지 않거나 삭제된 기능	20
1.2.11.3.1. Elasticsearch Curator가 더 이상 사용되지 않음	21
1.2.11.3.2. 레거시 Fluentd 및 레거시 syslog 방법을 사용하여 로그 전달이 더 이상 사용되지 않음	21
1.2.11.4. 버그 수정	21
1.2.11.5. 확인된 문제	23
<b>2장. RED HAT OPENSIFT LOGGING 이해</b> .....	<b>24</b>
2.1. OPENSIFT LOGGING 배포 이해	24

2.1.1. JSON OpenShift Container Platform 로깅 정보	24
2.1.2. Kubernetes 이벤트 수집 및 저장 정보	25
2.1.3. OpenShift Container Platform Logging 업데이트 정보	25
2.1.4. 클러스터 대시보드 보기 정보	25
2.1.5. OpenShift Container Platform 로깅 문제 해결 정보	25
2.1.6. OpenShift Container Platform 로깅 설치 제거 정보	25
2.1.7. 필드 내보내기 정보	25
2.1.8. OpenShift Logging 구성 요소 정보	26
2.1.9. 로깅 수집기 정보	26
2.1.10. 로그 저장소 정보	26
2.1.11. 로깅 시각화 정보	27
2.1.12. 이벤트 라우팅 정보	27
2.1.13. 로그 전송 정보	28
<b>3장. OPENSIFT LOGGING 설치</b>	<b>29</b>
3.1. 웹 콘솔을 사용하여 OPENSIFT LOGGING 설치	29
3.2. 설치 후 작업	34
3.3. CLI를 사용하여 OPENSIFT LOGGING 설치	34
3.4. 설치 후 작업	42
3.4.1. Kibana 인덱스 패턴 정의	42
3.4.2. 네트워크 분리가 활성화될 때 프로젝트 간 트래픽 허용	43
<b>4장. 로깅 배포 구성</b>	<b>45</b>
4.1. 클러스터 로깅 사용자 정의 리소스 정보	45
4.1.1. 클러스터 로깅 사용자 정의 리소스 정보	45
4.2. 로깅 수집기 구성	46
4.2.1. 지원되지 않는 구성 정보	46
4.2.2. 로깅 수집기 Pod 보기	46
4.2.3. 로그 수집기 CPU 및 메모리 제한 구성	47
4.2.4. 로그 전달자를 위한 고급 구성	48
4.2.5. 기본 Elasticsearch 로그 저장소를 사용하지 않는 경우 사용되지 않은 구성 요소 제거	51
4.3. 로그 저장소 구성	52
4.3.1. 감사 로그를 로그 저장소로 전달	53
4.3.2. 로그 보존 시간 구성	55
4.3.3. 로그 저장소에 대한 CPU 및 메모리 요청 구성	57
4.3.4. 로그 저장소에 대한 복제 정책 구성	58
4.3.5. Elasticsearch Pod 축소	59
4.3.6. 로그 저장소에 대한 영구 스토리지 구성	59
4.3.7. emptyDir 스토리지에 대한 로그 저장소 구성	60
4.3.8. Elasticsearch 롤링 클러스터 재시작 수행	61
4.3.9. 로그 저장소 서비스를 경로로 노출	64
4.4. 로깅 시각화 프로그램 구성	66
4.4.1. CPU 및 메모리 제한 구성	67
4.4.2. 로깅 시각화 프로그램 노드의 확장성 중복	68
4.5. OPENSIFT LOGGING 스토리지 구성	68
4.5.1. OpenShift Logging 및 OpenShift Container Platform에 대한 스토리지 고려 사항	68
4.5.2. 추가 리소스	69
4.6. OPENSIFT LOGGING 구성 요소에 대한 CPU 및 메모리 제한 구성	69
4.6.1. CPU 및 메모리 제한 구성	69
4.7. 허용 오차를 사용하여 OPENSIFT LOGGING POD 배치 제어	70
4.7.1. 허용 오차를 사용하여 로그 저장소 Pod 배치 제어	72
4.7.2. 허용 오차를 사용하여 로깅 시각화 프로그램 Pod 배치 제어	73
4.7.3. 허용 오차를 사용하여 로그 수집기 Pod 배치 제어	74

4.7.4. 추가 리소스	75
4.8. 노트 선택기로 OPENSIFT LOGGING 리소스 이동	75
4.8.1. OpenShift Logging 리소스 이동	75
4.9. SYSTEMD-JOURNALD 및 FLUENTD 구성	79
4.9.1. OpenShift Logging을 위한 systemd-journald 구성	79
4.10. 유지보수 및 지원	82
4.10.1. 지원되지 않는 구성 정보	82
4.10.2. 지원되지 않는 로깅 구성	82
4.10.3. 관리되지 않는 Operator에 대한 지원 정책	83
<b>5장. 리소스의 로그 보기</b>	<b>85</b>
5.1. 리소스 로그 보기	85
<b>6장. KIBANA를 사용하여 클러스터 로그 보기</b>	<b>87</b>
6.1. KIBANA 인덱스 패턴 정의	87
6.2. KIBANA에서 클러스터 로그 보기	88
<b>7장. 타사 시스템에 로그 전달</b>	<b>91</b>
7.1. 타사 시스템으로 로그 전달 정보	91
외부 로그 집계기를 사용할 수 없는 경우 Fluentd 로그 처리	95
7.2. 지원되는 로그 데이터 출력 유형	95
7.3. 외부 ELASTICSEARCH 인스턴스로 로그 전달	95
7.4. FLUENTD 정방향 프로토콜을 사용하여 로그 전달	98
7.5. SYSLOG 프로토콜을 사용하여 로그 전달	100
7.5.1. Syslog 매개변수	102
7.5.2. 추가 RFC5424 syslog 매개변수	103
7.6. KAFKA 브로커로 로그 전달	103
7.7. 특정 프로젝트의 애플리케이션 로그 전달	106
7.8. 특정 POD에서 애플리케이션 로그 전달	108
7.9. 레거시 FLUENTD 방법을 사용하여 로그 전달	110
7.10. 레거시 SYSLOG 방법을 사용하여 로그 전달	112
<b>8장. JSON 로깅 활성화</b>	<b>116</b>
8.1. JSON 로그 구문 분석	116
8.2. ELASTICSEARCH의 JSON 로그 데이터 구성	117
8.3. ELASTICSEARCH 로그 저장소로 JSON 로그 전달	119
<b>9장. 쿠버네티스 이벤트 수집 및 저장</b>	<b>121</b>
9.1. 이벤트 라우터 배포 및 구성	121
<b>10장. OPENSIFT LOGGING 업데이트</b>	<b>125</b>
10.1. OPENSIFT CONTAINER PLATFORM 4.6 또는 이전 버전의 OPENSIFT LOGGING 5.X로 클러스터 로깅에 서 업데이트	125
10.2. OPENSIFT LOGGING을 현재 버전으로 업데이트	129
<b>11장. 클러스터 대시보드 보기</b>	<b>134</b>
11.1. ELASTICSEARCH 및 OPENSIFT LOGGING 대시보드에 액세스	134
11.2. OPENSIFT 로깅 대시보드 정보	134
11.3. 로깅/ELASTICSEARCH 노트 대시보드의 차트	136
<b>12장. 로깅 문제 해결</b>	<b>141</b>
12.1. OPENSIFT LOGGING 상태 보기	141
12.1.1. Red Hat OpenShift Logging Operator의 상태 보기	141
12.1.1.1. 상태 메시지 예	143
12.1.2. OpenShift Logging 구성 요소의 상태 보기	145

12.2. 로그 저장소의 상태 보기	146
12.2.1. 로그 저장소의 상태 보기	146
12.2.1.1. 상태 메시지 예	148
12.2.2. 로그 저장소 구성 요소의 상태 보기	150
12.3. OPENSIFT LOGGING 경고 이해	154
12.3.1. 로깅 수집기 경고 보기	154
12.3.2. 로깅 수집기 경고 정보	154
12.3.3. Elasticsearch 경고 규칙 정보	155
12.4. RED HAT 지원을 위한 로깅 데이터 수집	156
12.4.1. must-gather 툴 정보	156
12.4.2. 사전 요구 사항	156
12.4.3. OpenShift Logging 데이터 수집	157
12.5. 심각한 경고 문제 해결	157
12.5.1. Elasticsearch 클러스터 상태가 빨간색임	157
12.5.2. Elasticsearch 클러스터 상태가 노란색임	159
12.5.3. Elasticsearch 노드 디스크 Low Watermark Reached	160
12.5.4. Elasticsearch 노드 디스크 High Watermark Reached	161
12.5.5. Elasticsearch 노드 디스크 Flood Watermark Reached	162
12.5.6. Elasticsearch JVM 힙 사용량이 높음	163
12.5.7. 집계된 로깅 시스템 CPU가 높음	164
12.5.8. Elasticsearch 프로세스 CPU가 높음	164
12.5.9. Elasticsearch 디스크 공간이 부족	164
12.5.10. Elasticsearch FileDescriptor 사용량이 높음	165
<b>13장. OPENSIFT LOGGING 설치 제거</b>	<b>166</b>
13.1. OPENSIFT CONTAINER PLATFORM에서 OPENSIFT LOGGING 설치 삭제	166
<b>14장. 로그 레코드 필드</b>	<b>169</b>
<b>15장. MESSAGE</b>	<b>170</b>
<b>16장. STRUCTURED</b>	<b>171</b>
<b>17장. @TIMESTAMP</b>	<b>172</b>
<b>18장. 호스트 이름</b>	<b>173</b>
<b>19장. IPADDR4</b>	<b>174</b>
<b>20장. IPADDR6</b>	<b>175</b>
<b>21장. LEVEL</b>	<b>176</b>
<b>22장. PID</b>	<b>177</b>
<b>23장. SERVICE</b>	<b>178</b>
<b>24장. TAGS</b>	<b>179</b>
<b>25장. FILE</b>	<b>180</b>
<b>26장. OFFSET</b>	<b>181</b>
<b>27장. KUBERNETES</b>	<b>182</b>
27.1. KUBERNETES.POD_NAME	182
27.2. KUBERNETES.POD_ID	182
27.3. KUBERNETES.NAMESPACE_NAME	182
27.4. KUBERNETES.NAMESPACE_ID	182

---

27.5. KUBERNETES.HOST	182
27.6. KUBERNETES.CONTAINER_NAME	182
27.7. KUBERNETES.ANNOTATIONS	183
27.8. KUBERNETES.LABELS	183
27.9. KUBERNETES.EVENT	183
27.9.1. kubernetes.event.verb	183
27.9.2. kubernetes.event.metadata	183
27.9.2.1. kubernetes.event.metadata.name	183
27.9.2.2. kubernetes.event.metadata.namespace	183
27.9.2.3. kubernetes.event.metadata.selfLink	184
27.9.2.4. kubernetes.event.metadata.uid	184
27.9.2.5. kubernetes.event.metadata.resourceVersion	184
27.9.3. kubernetes.event.involvedObject	184
27.9.3.1. kubernetes.event.involvedObject.kind	184
27.9.3.2. kubernetes.event.involvedObject.namespace	184
27.9.3.3. kubernetes.event.involvedObject.name	185
27.9.3.4. kubernetes.event.involvedObject.uid	185
27.9.3.5. kubernetes.event.involvedObject.apiVersion	185
27.9.3.6. kubernetes.event.involvedObject.resourceVersion	185
27.9.4. kubernetes.event.reason	185
27.9.5. kubernetes.event.source_component	186
27.9.6. kubernetes.event.firstTimestamp	186
27.9.7. kubernetes.event.count	186
27.9.8. kubernetes.event.type	186
<b>28장. OPENSIFT</b> .....	<b>187</b>
28.1. OPENSIFT.LABELS	187



# 1장. RED HAT OPENSIFT LOGGING 릴리스 노트

## 1.1. 보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 향후 여러 릴리스에 대해 단계적으로 구현될 예정입니다. 자세한 내용은 [Red Hat CTO Chris Wright의 메시지](#)에서 참조하십시오.

## 1.2. 지원되는 버전

표 1.1. Red Hat OpenShift Logging (RHOL)에 대한 OpenShift Container Platform 버전 지원

	4.7	4.8	4.9
RHOL 5.1	X	X	
RHOL 5.2	X	X	X
RHOL 5.3		X	X

### 1.2.1. OpenShift Logging 5.1.0

이번 릴리스에는 [RHSA-2021:2112 OpenShift Logging 버그 수정 릴리스 5.1.0](#)이 포함되어 있습니다.

#### 1.2.1.1. 새로운 기능 및 개선 사항

OpenShift Logging 5.1은 이제 다음에서 실행되는 OpenShift Container Platform 4.7 이상을 지원합니다.

- IBM Power Systems
- IBM Z 및 LinuxONE

이 릴리스에는 다음 구성 요소 및 개념과 관련된 개선 사항이 추가되었습니다.

- 클러스터 관리자는 Kubernetes Pod 레이블을 사용하여 애플리케이션에서 로그 데이터를 수집하여 특정 로그 수집기로 전달할 수 있습니다. **ClusterLogForwarder** CR(사용자 정의 리소스) YAML 파일에서 **inputs[].application.selector.matchLabels** 요소를 구성하여 로그 데이터를 수집할 수 있습니다. 수집된 로그 데이터를 네임스페이스로 필터링할 수도 있습니다. ([LOG-883](#))
- 이 릴리스에서는 다음과 같은 새로운 **ElasticsearchNodeDiskWatermarkReached** 경고가 OpenShift Elasticsearch Operator (EO)에 추가되었습니다.
  - Elasticsearch 노드 디스크 Low Watermark Reached
  - Elasticsearch 노드 디스크 High Watermark Reached
  - Elasticsearch 노드 디스크 Flood Watermark Reached

경고는 Elasticsearch 노드가 다음 6시간 내에 **Disk Low Watermark**, **Disk High Watermark**, **Disk Flood Stage Watermark** 차례로 임계값에 도달할 것으로 예측할 때 과거의 여러 경고를 적용합니다. 이 경고 기간을 사용하면 노드가 디스크 워터마크 임계값에 도달하기 전에 응답할 수

있습니다. 경고 메시지는 문제 해결 단계에 대한 링크도 제공하므로 문제를 완화하는 데 도움이 됩니다. EO는 지난 몇 시간 간의 디스크 공간 데이터를 선형 모델에 적용하여 이러한 경고를 생성합니다. (LOG-1100)

- JSON 로그는 이제 Red Hat의 관리 Elasticsearch 클러스터 또는 기타 지원되는 타사 시스템에 따옴표가 없는 문자열이 아니라 JSON 개체로 전달할 수 있습니다. 또한 Kibana 내부의 JSON 로그 메시지에서 개별 필드를 쿼리하여 특정 로그의 검색 가능성을 높일 수 있습니다. (LOG-785, LOG-1148)

### 1.2.1.2. 사용되지 않거나 삭제된 기능

이전 릴리스에서 사용 가능하던 일부 기능이 더 이상 사용되지 않거나 삭제되었습니다.

더 이상 사용되지 않는 기능은 여전히 OpenShift Logging에 포함되어 있으며 계속 지원됩니다. 그러나 이 기능은 향후 릴리스에서 제거될 예정이므로 새로운 배포에는 사용하지 않는 것이 좋습니다.

#### 1.2.1.2.1. Elasticsearch Curator가 제거됨

이번 업데이트에서는 Elasticsearch Curator가 제거되어 더 이상 지원되지 않습니다. Elasticsearch Curator는 OpenShift Container Platform 4.4 및 이전 버전에서 인덱스를 큐레이팅하거나 관리할 수 있도록 지원합니다. Elasticsearch Curator를 사용하는 대신 로그 보존 시간을 구성합니다.

#### 1.2.1.2.2. 레거시 Fluentd 및 레거시 syslog 방법을 사용하여 로그 전달이 더 이상 사용되지 않음

OpenShift Container Platform 4.6에서 현재로 레거시 Fluentd 및 레거시 syslog 방법을 사용하여 로그를 전달하지 않으며 향후 릴리스에서 제거됩니다. 대신 표준 비레거시 방법을 사용합니다.

### 1.2.1.3. 버그 수정

- 이 업데이트 이전에는 **ClusterLogForwarder** CR에 생성된 후 **input[].selector** 요소가 표시되지 않았습니다. 이번 업데이트를 통해 **ClusterLogForwarder** CR에서 **selector**를 지정할 때 계속 유지됩니다. 이 버그를 수정하려면 Pod 라벨 선택기를 사용하여 애플리케이션 로그 데이터를 전달할 수 있는 LOG-883에 필요합니다. (LOG-1338)
- 이 업데이트 이전에는 CSV(클러스터 서비스 버전) 업데이트에서 실수로 OpenShift Elasticsearch Operator 컨테이너의 리소스 및 제한이 잘못 설치되었습니다. 특정 조건에서 이로 인해 메모리 부족 상태가 발생하여 Elasticsearch Operator Pod가 종료되었습니다. 현재 릴리스에서는 Operator 컨테이너의 CSV 리소스 및 제한을 제거하여 이 문제가 해결되어 있습니다. perator가 문제 없이 예약됩니다. (LOG-1254)
- 이번 업데이트 이전에는 체인 인증서를 사용하여 Kafka로 로그를 전달하지 못했습니다. 다음 오류 메시지와 함께 실패했습니다.

**state=error: certificate verify failed (unable to get local issuer certificate)**

중간 CA에서 서명한 인증서를 사용하여 Kafka 브로커로 로그를 전달할 수 없었습니다. 이 문제는 fluentd Kafka 플러그인이 해당 시크릿의 **ca-bundle.crt** 항목에 제공된 단일 CA 인증서만 처리할 수 있었기 때문에 발생했습니다. 현재 릴리스에서는 fluentd Kafka 플러그인이 해당 시크릿의 **ca-bundle.crt** 항목에 제공된 여러 CA 인증서를 처리할 수 있도록 하여 이 문제를 해결합니다. 이제 중간 CA에서 서명한 인증서를 사용하여 Kafka 브로커로 로그를 전달할 수 있습니다. (LOG-1218, LOG-1216)

- 이번 업데이트 전에는 로드 중 Elasticsearch에서 클러스터에 문제가 없는 경우에도 HTTP 500 오류가 있는 일부 요청에 응답했습니다. 요청을 다시 시도하면 성공했습니다. 이 릴리스에서는 임시 HTTP 500 오류가 발생할 때 인덱스 관리 cron 작업을 업데이트하여 문제를 해결합니다. 업데이트된 인덱스 관리 cron 작업은 실패하기 전에 먼저 여러 번 요청을 재시도합니다. (LOG-1215)

- 이번 업데이트 이전에는 설치 구성에 **.proxy** 값을 설정하지 않은 다음 설치된 클러스터에 글로벌 프록시를 구성한 경우 버그로 인해 Fluentd가 로그를 Elasticsearch로 전달하지 못했습니다. 이 문제를 해결하려면 프록시 또는 클러스터 구성에서 **no\_proxy** 값을 **.svc.cluster.local**로 설정하여 내부 트래픽을 건너뛸니다. 이번 업데이트에서는 프록시 설정 문제가 수정되었습니다. 이제 OpenShift Container Platform 클러스터를 설치한 후 글로벌 프록시를 구성하면 Fluentd가 Elasticsearch로 로그를 전달합니다. ([LOG-1187](#), [BZ#1915448](#))
- 이번 업데이트 이전에는 로깅 수집기가 필요한 것보다 많은 소켓 연결을 생성했습니다. 이번 업데이트를 통해 로깅 수집기는 기존 소켓 연결을 사용하여 로그를 보냅니다. ([LOG-1186](#))
- 이번 업데이트 이전에는 클러스터 관리자가 Elasticsearch 클러스터에서 스토리지를 추가하거나 제거하려고 하면 OpenShift Elasticsearch Operator (EO)가 Elasticsearch 클러스터를 업그레이드하여 **scheduledUpgrade**를 표시하도록 잘못 시도했습니다. **"true", shardAllocationEnabled:** 기본 설정 및 볼륨을 변경합니다. 이번 업데이트에서는 EO에서 Elasticsearch 클러스터를 업그레이드하지 않습니다.  
EO 상태에는 지원되지 않는 변경을 무시한 Elasticsearch 스토리지 변경을 시도한 경우를 나타내는 다음과 같은 새 상태 정보가 표시됩니다.
  - 임시 스토리지 구조와 영구저장장치 구조를 사용하여 변경할 때 **StorageStructureChangelgnored**됩니다.
  - 스토리지 클래스 이름을 변경하려고 하면 **StorageClassNameChangelgnored**됩니다.
  - 스토리지 크기를 변경하려고 하면 **StorageSizeChangelgnored**됩니다.



### 참고

임시 스토리지에서 영구 스토리지로 전환하도록 **ClusterLogging** 사용자 정의 리소스(CR)를 구성하는 경우 EO는 PVC(영구 볼륨 클레임)를 생성하지만 PV(영구 볼륨)를 생성하지 않습니다. **StorageStructureChangelgnored** 상태를 지우려면 **ClusterLogging** CR로 변경 사항을 취소하고 PVC(영구 볼륨 클레임)를 삭제해야 합니다.

([LOG-1351](#))

- 이번 업데이트 이전에는 전체 Elasticsearch 클러스터를 재배포한 경우 하나의 비 데이터 노드가 실행되고 다른 모든 데이터 노드가 종료되는 비정상 상태가 되었습니다. 이는 새 인증서로 인해 Elasticsearch Operator가 Elasticsearch 클러스터의 데이터가 아닌 노드를 축소할 수 없었기 때문에 발생했습니다. 이번 업데이트를 통해 Elasticsearch Operator는 모든 데이터와 비 데이터 노드를 축소한 다음 다시 백업하여 새 인증서를 로드할 수 있습니다. Elasticsearch Operator는 새 인증서를 로드한 후 새 노드에 연결할 수 있습니다. ([LOG-1536](#))

## 1.2.2. OpenShift Logging 5.0.9

이 릴리스에는 [RHBA-2021:3705 - 버그 수정 권고가 포함되어 있습니다](#). OpenShift Logging 버그 수정 릴리스 (5.0.9).

### 1.2.2.1. 버그 수정

이 릴리스에는 다음과 같은 버그 수정이 포함되어 있습니다.

- 이번 업데이트 이전에는 일부 로그 항목에 인식할 수 없는 UTF-8 바이트가 있어 Elasticsearch가 메시지를 거부하고 전체 버퍼링된 페이로드를 차단했습니다. 이번 업데이트에서는 문제가 해결되었습니다. 거부된 페이로드는 유효하지 않은 로그 항목을 삭제하고 나머지 항목을 다시 제출합니다. ([LOG-1574](#))

- 이 번 업데이트 이전에는 **ClusterLogging** 사용자 정의 리소스(CR)를 편집해도 **totalLimitSize** 값이 Fluentd **total\_limit\_size** 필드에 적용되지 않아 버퍼 플러그인 인스턴스의 크기가 제한됩니다. 결과적으로 Fluentd가 기본값을 적용했습니다. 이번 업데이트를 통해 CR은 total **LimitSize** 값을 Fluentd **total\_limit\_size** 필드에 적용합니다. Fluentd는 **total\_limit\_size** 필드 또는 기본값 중 적은 값을 사용합니다. (LOG-1736)

### 1.2.2.2. CVE

- [CVE-2020-25648](#)
- [CVE-2021-22922](#)
- [CVE-2021-22923](#)
- [CVE-2021-22924](#)
- [CVE-2021-36222](#)
- [CVE-2021-37576](#)
- [CVE-2021-37750](#)
- [CVE-2021-38201](#)

### 1.2.3. OpenShift Logging 5.0.8

이 릴리스에는 RHBA-2021:3526 - 버그 수정 권고가 포함되어 있습니다. OpenShift Logging 버그 수정 릴리스 (5.0.8).

#### 1.2.3.1. 버그 수정

이 릴리스에는 다음과 같은 버그 수정이 포함됩니다.

- 릴리스 파이프라인 스크립트의 문제로 인해 **olm.skipRange** 필드의 값은 **5.2.0** 에서 변경되지 않고 z-stream 번호 **0** 이 증가하면 업데이트되지 않았습니다. 현재 릴리스에서는 릴리스 번호가 변경될 때 이 필드의 값을 업데이트하도록 파이프라인 스크립트를 수정합니다. (LOG-1741)

### 1.2.4. OpenShift Logging 5.0.7

이 릴리스에는 RHBA-2021:2884 - 버그 수정 권고가 포함되어 있습니다. OpenShift Logging 버그 수정 릴리스 (5.0.7).

#### 1.2.4.1. 버그 수정

이 릴리스에는 다음과 같은 버그 수정이 포함됩니다.

- [LOG-1594](#) - Vendored viaq/logerr 종속성이 라이선스 파일이 누락됨

#### 1.2.4.2. CVE

- [CVE-2016-10228](#)
- [CVE-2017-14502](#)
- [CVE-2018-25011](#)

- [CVE-2019-2708](#)
- [CVE-2019-3842](#)
- [CVE-2019-9169](#)
- [CVE-2019-13012](#)
- [CVE-2019-18276](#)
- [CVE-2019-18811](#)
- [CVE-2019-19523](#)
- [CVE-2019-19528](#)
- [CVE-2019-25013](#)
- [CVE-2020-0431](#)
- [CVE-2020-8231](#)
- [CVE-2020-8284](#)
- [CVE-2020-8285](#)
- [CVE-2020-8286](#)
- [CVE-2020-8927](#)
- [CVE-2020-9948](#)
- [CVE-2020-9951](#)
- [CVE-2020-9983](#)
- [CVE-2020-10543](#)
- [CVE-2020-10878](#)
- [CVE-2020-11608](#)
- [CVE-2020-12114](#)
- [CVE-2020-12362](#)
- [CVE-2020-12363](#)
- [CVE-2020-12364](#)
- [CVE-2020-12464](#)
- [CVE-2020-13434](#)
- [CVE-2020-13543](#)
- [CVE-2020-13584](#)

- [CVE-2020-13776](#)
- [CVE-2020-14314](#)
- [CVE-2020-14344](#)
- [CVE-2020-14345](#)
- [CVE-2020-14346](#)
- [CVE-2020-14347](#)
- [CVE-2020-14356](#)
- [CVE-2020-14360](#)
- [CVE-2020-14361](#)
- [CVE-2020-14362](#)
- [CVE-2020-14363](#)
- [CVE-2020-15358](#)
- [CVE-2020-15437](#)
- [CVE-2020-24394](#)
- [CVE-2020-24977](#)
- [CVE-2020-25212](#)
- [CVE-2020-25284](#)
- [CVE-2020-25285](#)
- [CVE-2020-25643](#)
- [CVE-2020-25704](#)
- [CVE-2020-25712](#)
- [CVE-2020-26116](#)
- [CVE-2020-26137](#)
- [CVE-2020-26541](#)
- [CVE-2020-27618](#)
- [CVE-2020-27619](#)
- [CVE-2020-27786](#)
- [CVE-2020-27835](#)
- [CVE-2020-28196](#)

- [CVE-2020-28974](#)
- [CVE-2020-29361](#)
- [CVE-2020-29362](#)
- [CVE-2020-29363](#)
- [CVE-2020-35508](#)
- [CVE-2020-36322](#)
- [CVE-2020-36328](#)
- [CVE-2020-36329](#)
- [CVE-2021-0342](#)
- [CVE-2021-0605](#)
- [CVE-2021-3177](#)
- [CVE-2021-3326](#)
- [CVE-2021-3501](#)
- [CVE-2021-3516](#)
- [CVE-2021-3517](#)
- [CVE-2021-3518](#)
- [CVE-2021-3520](#)
- [CVE-2021-3537](#)
- [CVE-2021-3541](#)
- [CVE-2021-3543](#)
- [CVE-2021-20271](#)
- [CVE-2021-23336](#)
- [CVE-2021-27219](#)
- [CVE-2021-33034](#)

## 1.2.5. OpenShift Logging 5.0.6

이 릴리스에는 [RHBA-2021:2655](#) - 버그 수정 권고가 포함되어 있습니다. OpenShift Logging 버그 수정 릴리스 (5.0.6).

### 1.2.5.1. 버그 수정

이 릴리스에는 다음과 같은 버그 수정이 포함됩니다.

- LOG-1451 - [1927249] fieldmanager.go:186] [SHOULD not HAPPEN] failed update managedFields...키에 대한 중복 항목 [name="POLICY\_MAPPING"]([LOG-1451](#))
- LOG-1537 - ES 클러스터에 비 데이터 노드가 포함되어 있으면 전체 클러스터 인증서 재배포가 중단됨([LOG-1537](#))
- LOG-1430 - 이벤트 라우터, "Observed a panic: &runtime.TypeAssertionError"([LOG-1430](#))
- LOG-1461 - 작업 로그에 오류가 있는 경우에도 인덱스 관리 작업 상태는 항상 **Completed**임 ([LOG-1461](#))
- LOG-1459 - Operator에 연결이 끊긴 주석이 없음 ([LOG-1459](#))
- LOG-1572 - 버그579: 모든 로그를 수집하도록 기본 제공 애플리케이션 동작 수정 ([LOG-1572](#))

### 1.2.5.2. CVE

- [CVE-2016-10228](#)
- [CVE-2017-14502](#)
- [CVE-2018-25011](#)
- [CVE-2019-2708](#)
- [CVE-2019-9169](#)
- [CVE-2019-25013](#)
- [CVE-2020-8231](#)
- [CVE-2020-8284](#)
- [CVE-2020-8285](#)
- [CVE-2020-8286](#)
- [CVE-2020-8927](#)
- [CVE-2020-10543](#)
- [CVE-2020-10878](#)
- [CVE-2020-13434](#)
- [CVE-2020-14344](#)
- [CVE-2020-14345](#)
- [CVE-2020-14346](#)
- [CVE-2020-14347](#)
- [CVE-2020-14360](#)
- [CVE-2020-14361](#)
- [CVE-2020-14362](#)

- [CVE-2020-14363](#)
- [CVE-2020-15358](#)
- [CVE-2020-25712](#)
- [CVE-2020-26116](#)
- [CVE-2020-26137](#)
- [CVE-2020-26541](#)
- [CVE-2020-27618](#)
- [CVE-2020-27619](#)
- [CVE-2020-28196](#)
- [CVE-2020-29361](#)
- [CVE-2020-29362](#)
- [CVE-2020-29363](#)
- [CVE-2020-36328](#)
- [CVE-2020-36329](#)
- [CVE-2021-3177](#)
- [CVE-2021-3326](#)
- [CVE-2021-3516](#)
- [CVE-2021-3517](#)
- [CVE-2021-3518](#)
- [CVE-2021-3520](#)
- [CVE-2021-3537](#)
- [CVE-2021-3541](#)
- [CVE-2021-20271](#)
- [CVE-2021-23336](#)
- [CVE-2021-27219](#)
- [CVE-2021-33034](#)

### 1.2.6. OpenShift Logging 5.0.5

이 릴리스에는 RHSA-2021:2374 - 보안 권고가 포함되어 있습니다. 보통: OpenShift Logging 버그 수정 릴리스 (5.0.5).

### 1.2.6.1. 보안 수정

- Gogo/protobuf: plugin/unmarshal/unmarshal.go에는 특정 인덱스 검증이 없습니다. ([CVE-2021-3121](#))
- glib: 64비트에서 32비트로 암시적으로 캐스팅되어 64비트 플랫폼에서 g\_bytes\_new 함수의 정수 오버플로입니다. ([CVE-2021-27219](#))

다음 문제는 위의 CVE와 관련되어 있습니다.

- BZ#1921650 gogo/protobuf: plugin/unmarshal/unmarshal.go에는 특정 인덱스 검증이 없습니다. ([BZ#1921650](#))
- LOG-1361 CVE-2021-3121 elasticsearch-operator-container: gogo/protobuf: plugin/unmarshal/unmarshal.go에는 특정 인덱스 검증[openshift-logging-5]이 없습니다. ([LOG-1361](#))
- LOG-1362 CVE-2021-3121 elasticsearch-proxy-container: gogo/protobuf: plugin/unmarshal/unmarshal.go에는 특정 인덱스 검증[openshift-logging-5]이 없습니다. ([LOG-1362](#))
- LOG-1363 CVE-2021-3121 logging-eventrouter-container: gogo/protobuf: plugin/unmarshal/unmarshal.go에는 특정 인덱스 검증[openshift-logging-5]이 없습니다. ([LOG-1363](#))

### 1.2.7. OpenShift Logging 5.0.4

이 릴리스에는 [RHSA-2021:2136](#) - 보안 권고가 포함되어 있습니다. 보통: [OpenShift Logging 보안 및 버그 업데이트 \(5.0.4\)](#).

#### 1.2.7.1. 보안 수정

- Gogo/protobuf: plugin/unmarshal/unmarshal.go에는 특정 인덱스 검증이 없습니다. ([CVE-2021-3121](#))

다음 Jira 문제에는 위의 CVE가 포함되어 있습니다.

- LOG-1364 CVE-2021-3121 cluster-logging-operator-container: gogo/protobuf: plugin/unmarshal/unmarshal.go에는 특정 인덱스 검증[openshift-logging-5]이 없습니다. ([LOG-1364](#))

#### 1.2.7.2. 버그 수정

이 릴리스에는 다음과 같은 버그 수정이 포함됩니다.

- LOG-1328 BZ-1945168의 5.0.z로 포트 수정 ([LOG-1328](#))

### 1.2.8. OpenShift Logging 5.0.3

이 릴리스에는 [RHSA-2021:1515](#) - 보안 권고가 포함되어 있습니다. [Important OpenShift Logging 버그 수정 릴리스 \(5.0.3\)](#).

#### 1.2.8.1. 보안 수정

- aust-databind: slf4j-ext 클래스에서 임의의 코드 실행 ([CVE-2018-14718](#))

- jackson-databind: blaze-ds-opt 및 blaze-ds-core 클래스에서 임의의 코드 실행 ([CVE-2018-14719](#))
- jackson-databind: 일부 JDK 클래스에서 exfiltration/XXE ([CVE-2018-14720](#))
- jackson-databind: axis2-jaxws 클래스에서 SSRF (server-side request forgery) ([CVE-2018-14721](#))
- jackson-databind: axis2-transport-jms 클래스에서 잘못된 다형성 직렬화 ([CVE-2018-19360](#))
- jackson-databind: openjpa 클래스에서 잘못된 다형성 직렬화 ([CVE-2018-19361](#))
- jackson-databind: jboss-common-core 클래스에서 잘못된 다형성 직렬화 ([CVE-2018-19362](#))
- jackson-databind: 기본 타이핑이 잘못되어 원격 코드가 실행 ([CVE-2019-14379](#))
- jackson-databind: com.pastdev.httpcomponents.configuration.JndiConfiguration에서 직렬화 가젯 ([CVE-2020-24750](#))
- jackson-databind: org.apache.commons.dbcp2.datasources.PerUserPoolDataSource와 관련하여 직렬화 가젯과 입력 사이의 상호 작용을 잘못 처리 ([CVE-2020-35490](#))
- jackson-databind: org.apache.commons.dbcp2.datasources.SharedPoolDataSource와 관련하여 직렬화 가젯과 입력 사이의 상호 작용을 잘못 처리 ([CVE-2020-35491](#))
- jackson-databind: com.oracle.wls.shaded.org.apache.xalan.lib.sql.JNDIConnectionPool와 관련하여 직렬화 가젯과 입력 사이의 상호 작용을 잘못 처리 ([CVE-2020-35728](#))
- jackson-databind: oadd.org.apache.commons.dbcp.cpdsadapter.DriverAdapterCPDS와 관련하여 직렬화 가젯과 입력 사이의 상호 작용을 잘못 처리 ([CVE-2020-36179](#))
- jackson-databind: org.apache.commons.dbcp2.cpdsadapter.DriverAdapterCPDS와 관련하여 직렬화 가젯과 입력 사이의 상호 작용을 잘못 처리 ([CVE-2020-36180](#))
- jackson-databind: org.apache.tomcat.dbcp.dbcp.cpdsadapter.DriverAdapterCPDS와 관련하여 직렬화 가젯과 입력 사이의 상호 작용을 잘못 처리 ([CVE-2020-36181](#))
- jackson-databind: org.apache.tomcat.dbcp.dbcp2.cpdsadapter.DriverAdapterCPDS와 관련하여 직렬화 가젯과 입력 사이의 상호 작용을 잘못 처리 ([CVE-2020-36182](#))
- jackson-databind: org.docx4j.org.apache.xalan.lib.sql.JNDIConnectionPool와 관련하여 직렬화 가젯과 입력 사이의 상호 작용을 잘못 처리 ([CVE-2020-36183](#))
- jackson-databind: org.apache.tomcat.dbcp.dbcp2.datasources.PerUserPoolDataSource와 관련하여 직렬화 가젯과 입력 사이의 상호 작용을 잘못 처리 ([CVE-2020-36184](#))
- jackson-databind: org.apache.tomcat.dbcp.dbcp2.datasources.SharedPoolDataSource와 관련하여 직렬화 가젯과 입력 사이의 상호 작용을 잘못 처리 ([CVE-2020-36185](#))
- jackson-databind: org.apache.tomcat.dbcp.dbcp.datasources.PerUserPoolDataSource와 관련하여 직렬화 가젯과 입력 사이의 상호 작용을 잘못 처리 ([CVE-2020-36186](#))
- jackson-databind: org.apache.tomcat.dbcp.dbcp.datasources.SharedPoolDataSource와 관련하여 직렬화 가젯과 입력 사이의 상호 작용을 잘못 처리 ([CVE-2020-36187](#))
- jackson-databind: com.newrelic.agent.deps.ch.qos.logback.core.db.JNDIConnectionSource와 관련하여 직렬화 가젯과 입력 사이의 상호 작용을 잘못 처리 ([CVE-2020-36188](#))

- jackson-databind: com.newrelic.agent.deps.ch.qos.logback.core.db.DriverManagerConnectionSource와 관련하여 직렬화 가젯과 입력 사이의 상호 작용을 잘못 처리 ([CVE-2020-36189](#))
- jackson-databind: javax.swing과 관련하여 직렬화 가젯과 입력 사이의 상호 작용을 잘못 처리 ([CVE-2021-20190](#))
- golang: ReverseProxy를 포함한 특정 net/http 서버의 데이터 경합이 DoS를 일으킬 수 있음 ([CVE-2020-15586](#))
- Golang: ReadUvarint 및 ReadVarint는 유효하지 않은 입력에서 무제한 바이트 수를 읽을 수 있습니다 ([CVE-2020-16845](#))
- OpenJDK: JAR 서명 비활성화 알고리즘의 불완전한 적용 (라이브러리, 8249906) ([CVE-2021-2163](#))

다음 Jira 문제에는 위의 CVE가 포함되어 있습니다.

- [LOG-1234](#) [CVE-2020-15586](#) [CVE-2020-16845](#) openshift-eventrouter: 각종 결함 [openshift-4]. ([LOG-1234](#))
- [LOG-1243](#) [CVE-2018-14718](#) [CVE-2018-14719](#) [CVE-2018-14720](#) [CVE-2018-19360](#) [CVE-2018-19361](#) [CVE-2019-14379](#) [CVE-2020-35490](#) [CVE-2020-35491](#) [CVE-2020-35728](#)... logging-elasticsearch6-container: 다양한 결함 [openshift-logging-5.0]. ([LOG-1243](#))

### 1.2.8.2. 버그 수정

이 릴리스에는 다음과 같은 버그 수정이 포함됩니다.

- [LOG-1224](#) 릴리스 5.0 - ClusterLogForwarder 네임 스페이스별 로그 전달이 예상대로 작동하지 않습니다. ([LOG-1224](#))
- [LOG-1232](#) 5.0 - Bug 1859004 - 이벤트 라우터가 이벤트 로그를 수집할 수 없는 경우가 있습니다. ([LOG-1232](#))
- [LOG-1299](#) 릴리스 5.0 - 체인된 인증서를 사용하여 Kafka에 로그를 전송하면 "state=error: certificate verify failed (local issuer certificate)" 오류와 함께 실패합니다. ([LOG-1299](#))

## 1.2.9. OpenShift Logging 5.0.2

이 릴리스에는 [RHBA-2021:1167](#) 버그 수정 권고가 포함되어 있습니다. [OpenShift Logging 버그 수정 릴리스 \(5.0.2\)](#).

### 1.2.9.1. 버그 수정

- 클러스터 설치 구성에 .proxy를 설정하지 않은 다음 설치된 클러스터에 글로벌 프록시를 구성한 경우 버그로 인해 Fluentd가 로그를 Elasticsearch로 전달하지 못했습니다. 이 문제를 해결하려면 proxy/cluster 구성에서 no\_proxy를 .svc.cluster.local로 설정하여 내부 트래픽을 건너뛸 수 있습니다. 현재 릴리스에서는 프록시 설정 문제가 수정되었습니다. 이제 OpenShift 클러스터를 설치한 후 글로벌 프록시를 구성하면 Fluentd가 Elasticsearch로 로그를 전달합니다. ([LOG-1187](#))
- 이전 버전에서는 체인 인증서를 사용하여 Kafka에 로그를 전달하지 못했습니다. "state=error: certificate verify failed (로컬 발급자 인증서를 가져올 수 없음)" 오류로 실패했습니다. 중간 CA에서 서명한 인증서를 사용하여 Kafka 브로커로 로그를 전달할 수 없었습니다. 이 문제는 fluentd Kafka 플러그인이 해당 시크릿의 ca-bundle.crt 항목에 제공된 단일 CA 인증서만 처리할 수 있기 때문에 발생했습니다. 현재 릴리스에서는 fluentd Kafka 플러그인이 해당 시크릿의 ca-

`bundle.crt` 항목에 제공된 여러 CA 인증서를 처리할 수 있도록 하여 이 문제를 해결합니다. 이제 중간 CA에서 서명한 인증서를 사용하여 Kafka 브로커로 로그를 전달할 수 있습니다. (LOG-1216, LOG-1218)

- 이전 버전에서는 CSV(클러스터 서비스 버전) 업데이트에서 실수로 OpenShift Elasticsearch Operator 컨테이너의 리소스 및 제한이 잘못 설치되었습니다. 특정 조건에서 이로 인해 메모리 부족 상태가 발생하여 Elasticsearch Operator Pod가 종료되었습니다. 현재 릴리스에서는 Operator 컨테이너의 CSV 리소스 및 제한을 제거하여 이 문제가 해결되어 있습니다. 이제 Operator가 문제 없이 예약됩니다. (LOG-1254)

## 1.2.10. OpenShift Logging 5.0.1

이 릴리스에는 RHBA-2021:0963 - 버그 수정 권고가 포함되어 있습니다. OpenShift Logging 버그 수정 릴리스 (5.0.1).

### 1.2.10.1. 버그 수정

- 이전 버전에서는 기존 로그 전달 시스템이 활성화된 경우 로그가 관리 스토리지로 전송되지 않았습니다. 생성된 로그 전달 구성이 현재 로그 전달 시스템 또는 기존 로그 전달 시스템 중에서 부적절하게 선택되었기 때문에 이 문제가 발생했습니다. 현재 릴리스에서는 이 문제가 해결되었습니다. ClusterLogging CR에서 logstore를 정의하는 경우 로그는 관리 스토리지로 전송됩니다. 또한 기존 로그 전달이 활성화된 경우 관리 스토리지가 활성화되었는지 여부에 관계없이 로그가 기존 로그 전달으로 전송됩니다. (LOG-1172)
- 이전 버전에서는 로드 중 Elasticsearch에서 클러스터에 문제가 없는 경우에도 HTTP 500 오류가 있는 일부 요청에 응답했습니다. 요청을 다시 시도하면 성공했습니다. 이 릴리스에서는 임시 HTTP 500 오류가 발생할 때 cron 작업을 업데이트하여 문제를 해결합니다. 이제 실패하기 전에 여러 번 요청을 다시 시도합니다. (LOG-1215)

## 1.2.11. OpenShift Logging 5.0.0

이 릴리스에는 RHBA-2021:0652 - 버그 수정 권고가 포함되어 있습니다. Errata Advisory for OpenShift Logging 5.0.0.

### 1.2.11.1. 새로운 기능 및 개선 사항

이 릴리스에는 다음 개념과 관련된 개선 사항이 추가되었습니다.

클러스터 로깅이 Red Hat OpenShift Logging이 됨  
이번 릴리스에서는 클러스터 로깅이 Red Hat OpenShift Logging 5.0이 됩니다.

인덱스당 최대 5개의 기본 샤드

이번 릴리스에서 OpenShift Elasticsearch Operator (EO)는 클러스터에 정의된 데이터 노드 수에 따라 1~5 사이에서 인덱스의 기본 샤드 수를 설정합니다.

이전에는 EO에서 인덱스의 샤드 수를 데이터 노드 수로 설정했습니다. Elasticsearch의 인덱스가 여러 복제본으로 설정된 경우 인덱스가 아닌 각 기본 샤드 마다 다수의 복제본을 만듭니다. 따라서 인덱스 샤드화로 클러스터에서 더 많은 복제본 샤드가 존재하게 되고 이로 인해 클러스터가 복제 및 동기화될 수 있는 오버헤드가 많이 생성되었습니다.

OpenShift Elasticsearch Operator 이름 및 수준 업데이트

이번 릴리스에서는 OpenShift Elasticsearch Operator의 표시 이름 및 Operator의 수준이 업데이트되었습니다. OpenShift Elasticsearch Operator의 새로운 표시 이름 및 특정 용도가 Operator Hub에서 업데이트되어 있습니다.

**OpenShift Elasticsearch Operator에서 CSV 성공 사례 보고**

이번 릴리스에서는 OpenShift Elasticsearch Operator에 대한 **ClusterServiceVersion (CSV)** 오브젝트를 설치하거나 업그레이드했음을 나타내는 보고 메트릭이 추가되었습니다. 이전 버전에서는 CSV를 설치 또는 업그레이드에 실패한 경우 경고를 확인하거나 생성할 수 있는 방법이 없었습니다. 이제 OpenShift Elasticsearch Operator의 일부로 경고가 제공됩니다.

**Elasticsearch pod 인증서 권한 경고 감소**

이전 버전에서는 Elasticsearch pod를 시작할 때 인증서 권한 경고가 생성되어 클러스터의 문제 해결에 있어서 일부 사용자를 잘못 유도했습니다. 현재 릴리스에서는 이러한 권한 문제를 수정하여 이러한 유형의 경고가 감소되었습니다.

**경고에서 설명 및 문제 해결에 새로운 링크**

이번 릴리스에서는 Elasticsearch 클러스터가 생성되는 경고에서 해당 경고에 대한 설명 및 문제 해결 단계의 페이지에 링크가 추가되었습니다.

**삭제 작업에 대한 새로운 연결 시간 제한**

현재 릴리스에서는 삭제 작업에 대한 연결 시간 제한이 추가되어 Pod가 인덱스를 삭제하기 위해 Elasticsearch를 쿼리할 때 Pod가 중단되는 것을 방지합니다. 이제 시간이 경과하기 전에 기본 'curl' 호출이 연결되지 않으면 시간 초과로 호출이 종료됩니다.

**롤오버 인덱스 템플릿에 업데이트를 최소화**

이번 업데이트에서는 OpenShift Elasticsearch Operator 필드 값이 다른 경우에만 롤오버 인덱스 템플릿을 업데이트합니다. 인덱스 템플릿은 인덱스보다 우선 순위가 높습니다. 템플릿이 업데이트되면 클러스터가 인덱스 샤드를 통해 배포 우선 순위를 지정하여 성능에 영향을 미칩니다. Elasticsearch 클러스터 작업을 최소화하기 위해 Operator는 현재 구성된 기본 샤드 또는 복제본 샤드 수가 변경될 경우에만 템플릿을 업데이트합니다.

**1.2.11.2. 기술 프리뷰 기능**

이 릴리스의 일부 기능은 현재 기술 프리뷰 단계에 있습니다. 이러한 실험적 기능은 프로덕션용이 아닙니다. 해당 기능은 Red Hat Customer Portal의 지원 범위를 참조하십시오.

**기술 프리뷰 기능 지원 범위**

아래 표에서 기능은 다음 상태로 표시됩니다.

- **TP: 기술 프리뷰**
- **GA: 정식 출시일 (GA)**
- **-: 사용할 수 없음**

표 1.2. 기술 프리뷰

기능	OCP 4.5	OCP 4.6	Logging 5.0
로그 전송	TP	GA	GA

**1.2.11.3. 사용되지 않거나 삭제된 기능**

이전 릴리스에서 사용 가능하던 일부 기능이 더 이상 사용되지 않거나 삭제되었습니다.

더 이상 사용되지 않는 기능은 여전히 OpenShift Logging에 포함되어 있으며 계속 지원됩니다. 그러나 이 기능은 향후 릴리스에서 제거될 예정이므로 새로운 배포에는 사용하지 않는 것이 좋습니다.

### 1.2.11.3.1. Elasticsearch Curator가 더 이상 사용되지 않음

Elasticsearch Curator는 더 이상 사용되지 않으며 향후 릴리스에서 제거됩니다. Elasticsearch Curator는 OpenShift Container Platform 4.4 및 이전 버전에서 인덱스를 큐레이팅하거나 관리할 수 있도록 지원합니다. Elasticsearch Curator를 사용하는 대신 로그 보존 시간을 구성합니다.

### 1.2.11.3.2. 레거시 Fluentd 및 레거시 syslog 방법을 사용하여 로그 전달이 더 이상 사용되지 않음

OpenShift Container Platform 4.6에서 현재로 레거시 Fluentd 및 레거시 syslog 방법을 사용하여 로그를 전달하지 않으며 향후 릴리스에서 제거됩니다. 대신 표준 비레거시 방법을 사용합니다.

### 1.2.11.4. 버그 수정

- 이전 버전에서는 Elasticsearch는 헤더가 기본 최대 헤더 크기인 8KB를 초과한 HTTP 요청을 거부했습니다. 이제 최대 헤더 크기는 128 KB이고 Elasticsearch는 더 이상 최대 헤더 크기를 초과한 HTTP 요청을 거부하지 않습니다. (BZ#1845293)
- 이전 버전에서는 소프트웨어 버그가 Elasticsearch 사용자 정의 리소스(CR)에서 노드의 상태를 올바르게 업데이트하지 않았기 때문에 노드가 **Pending** 상태에서 복구되지 않았습니다. 현재 릴리스에서는 이 문제가 해결되어 노드가 **Pending** 상태일 때 노드를 복구할 수 있습니다. (BZ#1887357)
- 이전 버전에서는 CLO(Cluster Logging Operator)에서 **clusterlogging** CR의 Elasticsearch 노드 수를 3개의 노드로 축소할 때 고유 ID가 있는 이전에 생성된 노드를 생략했습니다. OpenShift Elasticsearch Operator는 고유 ID가 있는 노드가 제거되지 않도록 하는 보호 조치가 있었기 때문에 업데이트를 거부했습니다. 이제 CLO가 노드 수를 축소하고 Elasticsearch CR을 업데이트할 때 할당하지 않고 고유한 ID가 있는 노드를 카운트 0으로 표시합니다. 결과적으로 **clusterlogging** CR을 사용하여 클러스터를 3개의 노드로 축소할 수 있습니다. (BZ#1879150)



#### 참고

OpenShift Logging 5.0 이상에서 Cluster Logging Operator는 Red Hat OpenShift Logging Operator라고 합니다.

- 이전 버전에서는 **ClusterLogForwarder**에서 시크릿이 잘못 설정되어 있으면, **Fluentd** 수집기 Pod 크래시 루프에 진입했습니다. 현재 릴리스에서는 이 문제가 해결되었습니다. 이제 **ClusterLogForwarder**에서 시크릿을 검증하고 상태 필드에 오류를 보고합니다. 결과적으로 **Fluentd** 수집기 Pod가 충돌하지 않습니다. (BZ#1888943)
- 이전 버전에서는 **clusterlogging** 인스턴스의 Kibana 리소스 설정을 **resource {}**에 테이트하면 생성되는 nil 맵에 패닉을 발생시켜 OpenShift Elasticsearch Operator의 상태가 **CrashLoopBackOff**로 변경되었습니다. 현재 릴리스는 맵을 초기화하여 이 문제를 해결되어 있습니다. (BZ#1889573)
- 이전 버전에서는 **ClusterLogForwarder**에서 동일한 암호를 사용하여 여러 출력을 생성할 때 **Fluentd** 수집기 Pod가 크래시 루프에 진입했습니다. 현재 릴리스에서는 이 문제가 해결되었습니다. 이제 여러 출력이 시크릿을 공유할 수 있습니다. (BZ#1890072)
- 이전 버전에서는 Kibana 경로를 삭제한 경우 CO(Cluster Logging Operator)를 복구하거나 재생성할 수 없었습니다. 이제 CLO에서 경로를 모니터링하고 경로를 삭제하면 OpenShift Elasticsearch Operator가 이를 조정하거나 다시 생성할 수 있습니다. (BZ#1890825)
- 이전 버전에서는 CLO(Cluster Logging Operator)에서 Red Hat이 제공하는 Elastic CRD(Elastic Custom Resource Definition)에 종속된 Elasticsearch 리소스를 조정하려고 했습니다. CLO에서 알 수 없는 유형의 목록을 나열하려고 하면 CLO가 조정 루프를 종료했습니다. 이는 CLO가 정의된지 여부에 관계없이 모든 관리 리소스를 조정하려고 시도했기 때문입니다. 현재 릴리스

스에서는 이 문제가 해결되었습니다. 사용자가 관리 스토리지를 정의하는 경우 CLO는 OpenShift Elasticsearch Operator에서 제공하는 유형만 조정합니다. 따라서 사용자는 CLO를 배포하여 클러스터 로깅의 수집기 전용 배포를 생성할 수 있습니다. (BZ#1891738)

- 이전 버전에서는 RFC 3164의 LF GA syslog 구현으로 인해 원격 syslog로 전송된 로그가 기존 동작과 호환되지 않았습니다. 현재 릴리스에서는 이 문제가 해결되었습니다. AddLogSource는 로그 소스의 세부 정보를 "message" 필드에 추가합니다. 이제 원격 syslog로 전송된 로그는 기존 동작과 호환됩니다. (BZ#1891886)
- 이전 버전에서는 Elasticsearch 롤오버 Pod가 resource\_already\_exists\_exception 오류로 실패했습니다. Elasticsearch 롤오버 API 내에서 다음 인덱스가 생성될 때 \* - write 별칭이 이를 참조하도록 업데이트되지 않았습니다. 그 결과 다음에 해당 특정 인덱스에 대해 롤오버 API 엔드포인트가 트리거되면 리소스가 이미 존재하는 오류를 수신했습니다.  
현재 릴리스에서는 이 문제가 해결되었습니다. 이제 indexmanagement cronjob에서 롤오버가 발생할 때 새 인덱스가 생성되면 별칭이 새 인덱스를 가리키는지 확인합니다. 이러한 동작은 오류 발생을 방지합니다. 클러스터가 이미 이 오류를 수신하는 경우 cronjob에서 문제가 해결되어 후속 실행이 예상대로 작동되도록 합니다. 이제 롤오버를 수행해도 더 이상 예외가 발생하지 않습니다. (BZ#1893992)
- 이전 버전에서는 로깅 스택이 작동하는 경우에도 Fluent가 로그 전송을 중지했습니다. 엔드포인트가 백업된 경우에도 로그는 장기간 엔드포인트에 전송되지 않았습니다. 이는 최대 백 오프 시간이 너무 길고 엔드포인트가 중지된 경우 발생했습니다. 현재 릴리스에서는 최대 백 오프 시간을 단축하여 이 문제를 해결하므로 더 빨리 로그가 발송됩니다. (BZ#1894634)
- 이전 버전에서는 Elasticsearch 노드의 스토리지 크기를 생략하면 OpenShift Elasticsearch Operator 코드에서 패닉 상태가 발생했습니다. 이 패닉은 로그에 다음과 같이 표시됩니다. 패닉을 감지했습니다. "잘못된 메모리 주소 또는 nil 포인터 역참조" 패닉은 스토리지 크기가 필수 필드이지만 소프트웨어가 이를 확인하지 않았기 때문에 발생했습니다. 현재 릴리스에서는 이 문제가 해결되어 스토리지 크기가 생략되어도 문제가 발생하지 않습니다. 대신 스토리지는 기본적으로 임시 스토리지로 설정되어 사용자에게 로그 메시지를 생성합니다. (BZ#1899589)
- 이전 버전에서는 elasticsearch-rollover 및 elasticsearch-delete Pod가 Invalid JSON: 또는 ValueError에 있었습니다. JSON 객체를 디코딩할 수 없습니다 오류 상태. 잘못된 JSON 입력에 대한 예외 처리가 없기 때문에 이러한 예외가 발생했습니다. 현재 릴리스에서는 잘못된 JSON 입력에 대한 처리를 제공하여 이 문제를 해결합니다. 그 결과 프로세스는 예외 추적 대신 오류 메시지를 출력하고 elasticsearch-rollover 및 elasticsearch-delete 작업은 이러한 오류 상태를 유지하지 않습니다. (BZ#1899905)
- 이전 버전에서는 Fluentd를 독립형으로 배포할 때 replicas 값이 0인 경우에도 Kibana pod가 생성되었습니다. 이 문제는 Elasticsearch 노드가 없는 경우에도 Kibana가 기본적으로 1 개의 Pod로 설정되었기 때문입니다. 현재 릴리스에서는 이 문제가 해결되었습니다. 이제 Kibana가 하나 이상의 Elasticsearch 노드가 있는 경우에만 1로 기본 설정됩니다. (BZ#1901424)
- 이전 버전에서는 시크릿을 삭제한 경우 다시 생성되지 않았습니다. 인증서가 Operator의 로컬에 있지만 변경되지 않아 다시 작성되지 않았습니다. 즉 인증서가 변경된 경우에만 작성되었습니다. 현재 릴리스에서는 이 문제가 해결되었습니다. 인증서가 변경되거나 찾을 수 없는 경우 시크릿이 다시 작성됩니다. 이제 마스터 인증서를 삭제하면 대체됩니다. (BZ#1901869)
- 이전 버전에서는 클러스터에 이름이 동일한 여러 사용자 지정 리소스가 있는 경우 API 그룹으로 정규화되어 있지 않았을 때 리소스가 알파벳순으로 선택되었습니다. 결과적으로 OpenShift Elastic Elasticsearch Operator와 함께 Red Hat의 OpenShift Elasticsearch Operator를 모두 설치하는 경우 must-gather 보고서를 통해 데이터를 수집할 때 오류가 발생했습니다. 현재 릴리스에서는 must-gathers가 클러스터의 사용자 정의 리소스에 대한 정보를 수집할 때 전체 API 그룹을 사용하도록 설정하여 이 문제가 해결되어 있습니다. (BZ#1897731)
- 인증서 생성과 관련된 문제를 해결하기 위한 이전 버그 수정으로 오류가 발생했습니다. 인증서를 읽으려면 인증서가 누락된 것으로 인식되었기 때문에 인증서가 다시 생성되었습니다. 이로 인해

OpenShift Elasticsearch Operator가 Elasticsearch 클러스터에서 롤링 업그레이드를 수행하여 잠재적으로 일치하지 않는 인증서가 생성되었습니다. 이 버그는 Operator에서 작업 디렉터리에 인증서를 잘못 작성하여 발생했습니다. 현재 릴리스에서는 이 문제가 해결되었습니다. 이제 Operator가 지속적으로 동일한 작업 디렉터리에 인증서를 읽기 및 쓰기를 실행하므로 필요한 경우 인증서를 다시 생성할 수 있습니다. (BZ#1905910)

- 이전 버전에서는 Elasticsearch 버전을 검색하기 위해 root 엔드 포인트에 대한 쿼리를 실행하면 403 응답이 수신되었습니다. 403 응답으로 인해 이전 릴리스에서 이 엔드포인트를 사용한 서비스가 중단되었습니다. 이 오류는 관리자 이외의 사용자에게 root 엔드 포인트의 쿼리 및 Elasticsearch 버전을 검색하는 데 필요한 **monitor** 권한이 없기 때문에 발생했습니다. 이제 관리자 이외의 사용자는 배포된 Elasticsearch 버전에 대해 root 엔드포인트를 쿼리할 수 있습니다. (BZ#1906765)
- 이전 버전에서는 일부 대규모 삽입 상황에서 Elasticsearch 프록시가 fluentd와 Elasticsearch 간의 연결 시간이 초과되었습니다. 그 결과 fluentd가 메시지를 전달하지 못하고 서버는 **Server returned nothing (no headers, no data)** 오류를 반환했습니다. 현재 릴리스에서는 이 문제가 해결되었습니다. Elasticsearch 프록시의 기본 HTTP 읽기 및 쓰기 시간 초과가 5초에서 1분으로 늘어납니다. 또한 필드의 HTTP 시간 초과를 제어하는 Elasticsearch 프록시의 명령줄 옵션도 제공됩니다. (BZ#1908707)
- 이전 버전에서는 대시보드 설정 리소스가 다른 네임스페이스 소유자를 참조하여 OpenShift Container Platform이 해당 리소스를 수집하도록 했기 때문에 {ProductName}/Elasticsearch 대시보드가 OpenShift Container Platform 모니터링 대시보드에서 누락되어 있는 경우도 있었습니다. 소유자의 참조는 OpenShift Elasticsearch Operator 조정 기능 (reconciler) 설정에서 삭제되고 로깅 대시 보드가 콘솔에 표시됩니다. (BZ#1910259)
- 이전 버전에서는 환경 변수를 사용하여 Kibana 구성 파일의 값을 대체하는 코드에서 주석 처리된 행을 고려하지 않았습니다. 이로 인해 사용자가 `server.maxPayloadBytes`의 기본값을 덮어쓰지 못했습니다. 현재 릴리스에서는 `server.maxPayloadByteswithin`의 기본값 코멘트를 해제하여 이 문제가 해결되었습니다. 사용자는 설명된 대로 환경 변수를 사용하여 값을 재정의할 수 있습니다. (BZ#1918876)
- 이전 버전에서는 마이그레이션에 실패한 인덱스를 삭제하기 위한 명령이 억제되지 않도록 Kibana 로스 수준을 높였고 이로 인해 Kibana 사용자의 이메일 주소 및 OAuth 토큰이 포함된 INFO 수준에서 GET 요청이 표시되었습니다. 현재 릴리스에서는 이러한 필드를 마스킹하여 이 문제를 해결하였으므로 Kibana 로그는 이 필드를 표시하지 않습니다. (BZ#1925081)

#### 1.2.11.5. 확인된 문제

- **ruby-kafka-1.1.0** 및 **fluent-plugin-kafka-0.13.1** gems가 있는 Fluentd Pod는 Apache Kafka 버전 0.10.1.0과 호환되지 않습니다. 결과적으로 Kafka에 로그 전송이 실패하고 **error\_class=Kafka::DeliveryFailed error="Failed to send messages to flux-openshift-v4/1"**의 메시지가 표시됩니다.

**ruby-kafka-0.7** gem은 Kafka 0.11의 네이티브 지원을 우선하여 Kafka 0.10에 대한 지원을 중단했습니다. **ruby-kafka-1.0.0** gem에 Kafka 2.3 및 2.4 지원이 추가되었습니다. 현재 OpenShift Logging은 Kafka 버전 2.4.1을 테스트하고 이를 지원합니다.

이 문제를 해결하려면 지원되는 Apache Kafka 버전으로 업그레이드합니다.

(BZ#1907370)

## 2장. RED HAT OPENSIFT LOGGING 이해

클러스터 관리자는 OpenShift Logging을 배포하여 OpenShift Container Platform 클러스터의 모든 로그(예: 노드 시스템 감사 로그, 애플리케이션 컨테이너 로그 및 인프라 로그)를 집계할 수 있습니다. OpenShift Logging은 클러스터 전체에서 이러한 로그를 집계하여 기본 로그 저장소에 저장합니다. [Kibana 웹 콘솔을 사용하여 로그 데이터를 시각화](#)할 수 있습니다.

OpenShift Logging에서는 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 [로그 저장소에 감사 로그 전송](#)에 설명된 대로 로그 전달 API를 사용해야 합니다.

### 2.1. OPENSIFT LOGGING 배포 이해

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

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

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

자세한 내용은 [로그 수집기 구성](#)을 참조하십시오.

#### 2.1.1. 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.

### 2.1.2. 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 저장](#)을 참조하십시오.

### 2.1.3. 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.

### 2.1.4. 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.

### 2.1.5. 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

### 2.1.6. OpenShift Container Platform 로깅 설치 제거 정보

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

자세한 내용은 [OpenShift Container Platform 로깅 설치 제거](#)를 참조하십시오.

### 2.1.7. 필드 내보내기 정보

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

자세한 내용은 [필드 내보내기](#) 정보를 참조하십시오.

## 2.1.8. OpenShift Logging 구성 요소 정보

OpenShift Logging 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 고급 시각화 및 대시보드를 생성할 수 있습니다.

OpenShift Logging의 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.

## 2.1.9. 로깅 수집기 정보

OpenShift Container Platform은 Fluentd를 사용하여 컨테이너 및 노드 로그를 수집합니다.

기본적으로 로그 수집기는 다음 소스를 사용합니다.

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

감사 로그를 수집하기 위해 로그 수집기를 구성하면 `/var/log/audit/audit.log`에서 해당 로그를 가져옵니다.

로깅 수집기는 데몬 세트로 각 OpenShift Container Platform 노드에 Pod를 배포합니다. 시스템 및 인프라 로그는 `journal`가 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 노력으로 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 구성](#)을 참조하십시오.

## 2.1.10. 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 API**를 사용하여 로그 전송 기능으로 로그를 외부 로그 저장소로 전송할 수 있습니다.

OpenShift Logging 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 인덱스로 구성된 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.

Red Hat OpenShift Logging Operator 및 그에 동반되는 OpenShift Elasticsearch Operator는 각 Elasticsearch 노드가 자체 스토리지 볼륨이 있는 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 설명서](#)를 참조하십시오.



#### 참고

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

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

자세한 내용은 [로그 저장소 구성](#)을 참조하십시오.

### 2.1.11. 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 구성](#)을 참조하십시오.

### 2.1.12. 이벤트 라우팅 정보

이벤트 라우터는 OpenShift Logging으로 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 저장](#)을 참조하십시오.

### 2.1.13. 로그 전송 정보

기본적으로 OpenShift Logging은 **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.

## 3장. OPENSIFT LOGGING 설치

OpenShift Elasticsearch 및 Red Hat OpenShift Logging Operator를 배포하여 OpenShift Logging을 설치할 수 있습니다. OpenShift Elasticsearch Operator는 OpenShift Logging에 사용되는 Elasticsearch 클러스터를 생성하고 관리합니다. Red Hat OpenShift Logging Operator는 로깅 스택의 구성 요소를 생성하고 관리합니다.

OpenShift Container Platform에 OpenShift Logging을 배포하는 프로세스에는 다음이 포함됩니다.

- [OpenShift Logging 스토리지 고려 사항 검토](#).
- OpenShift Container Platform 웹 콘솔 또는 CLI를 사용하여 OpenShift Elasticsearch Operator 및 Red Hat OpenShift Logging Operator 설치

### 3.1. 웹 콘솔을 사용하여 OPENSIFT LOGGING 설치

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



#### 참고

즉, 기본 Elasticsearch 로그 저장소를 사용하지 않는 경우 **ClusterLogging** 사용자 정의 리소스(CR)에서 내부 Elasticsearch **logStore**, Kibana **visualization** 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 제거](#)를 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 3개의 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.

#### 프로세스

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

1. OpenShift Elasticsearch Operator를 설치합니다.
  - a. OpenShift Container Platform 웹 콘솔에서 Operator → OperatorHub를 클릭합니다.
  - b. 사용 가능한 Operator 목록에서 OpenShift Elasticsearch Operator를 선택한 다음 설치를 클릭합니다.

- c. 설치 모드에서 클러스터의 모든 네임스페이스가 선택되어 있는지 확인합니다.
  - d. 설치된 네임스페이스에서 `openshift-operators-redhat`이 선택되어 있는지 확인합니다.  
**openshift-operators-redhat** 네임스페이스를 지정해야 합니다. **openshift-operators** 네임스페이스에 신뢰할 수 없는 **Community Operator**가 포함될 수 있고, 여기에서 **OpenShift Container Platform** 지표와 동일한 이름의 지표를 게시하면 충돌이 발생합니다.
  - e. 이 네임스페이스에서 **Operator** 권장 클러스터 모니터링 사용을 선택합니다.  
이 옵션은 네임스페이스 오브젝트에서 `openshift.io/cluster-monitoring: "true"` 레이블을 설정합니다. 클러스터 모니터링이 **openshift-operators-redhat** 네임스페이스를 스크랩하도록 하려면 이 옵션을 선택해야 합니다.
  - f. `stable-5.x`을 업데이트 채널로 선택합니다.
  - g. 승인 전략을 선택합니다.
    - 자동 전략을 사용하면 **Operator** 새 버전이 준비될 때 **OLM(Operator Lifecycle Manager)**이 자동으로 **Operator**를 업데이트할 수 있습니다.
    - 수동 전략을 사용하려면 적절한 자격 증명을 가진 사용자가 **Operator** 업데이트를 승인해야 합니다.
  - h. 설치를 클릭합니다.
  - i. **Operator** → 설치된 **Operator** 페이지로 전환하여 **OpenShift Elasticsearch Operator**가 설치되었는지 확인합니다.
  - j. 상태가 성공인 모든 프로젝트에 **OpenShift Elasticsearch Operator**가 나열되어 있는지 확인합니다.
2. **Red Hat OpenShift Logging Operator**를 설치합니다.
    - a. **OpenShift Container Platform** 웹 콘솔에서 **Operator** → **OperatorHub**를 클릭합니다.
    - b. 사용 가능한 **Operator** 목록에서 **Red Hat OpenShift Logging**을 선택한 다음 설치를 클릭합니다.
    - c. 설치 모드에서 클러스터의 특정 네임스페이스가 선택되어 있는지 확인합니다.
    - d. 설치된 네임스페이스에서 **Operator** 권장 네임스페이스가 `openshift-logging`인지 확인하십시오.
    - e. 이 네임스페이스에서 **Operator** 권장 클러스터 모니터링 사용을 선택합니다.  
이 옵션은 네임스페이스 오브젝트에서 `openshift.io/cluster-monitoring: "true"` 레이블을 설정합니다. 클러스터 모니터링이 `openshift-logging` 네임스페이스를 스크랩하도록 하려면 이 옵션을 선택해야 합니다.
    - f. `stable-5.x`을 업데이트 채널로 선택합니다.
    - g. 승인 전략을 선택합니다.
      - 자동 전략을 사용하면 **Operator** 새 버전이 준비될 때 **OLM(Operator Lifecycle Manager)**이 자동으로 **Operator**를 업데이트할 수 있습니다.
      - 수동 전략을 사용하려면 적절한 자격 증명을 가진 사용자가 **Operator** 업데이트를 승인해야 합니다.
    - h. 설치를 클릭합니다.

- i. Operator → 설치된 Operator 페이지로 전환하여 Red Hat OpenShift Logging Operator가 설치되었는지 확인합니다.
- j. Red Hat OpenShift Logging이 openshift-logging 프로젝트에 성공 상태로 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. OpenShift Logging 인스턴스를 생성합니다.

- a. 관리 → 사용자 정의 리소스 정의 페이지로 전환합니다.
- b. 사용자 정의 리소스 정의 페이지에서 ClusterLogging을 클릭합니다.
- c. 사용자 정의 리소스 정의 상세 정보 페이지의 작업 메뉴에서 인스턴스 보기를 선택합니다.
- d. ClusterLoggings 페이지에서 ClusterLogging 생성을 클릭합니다.  
데이터를 로드하기 위해 페이지를 새로 고쳐야 할 수도 있습니다.
- e. YAML 필드에서 코드를 다음으로 교체합니다.



#### 참고

이 기본 OpenShift Logging 구성은 다양한 환경을 지원해야 합니다. OpenShift Logging 클러스터에 수행할 수 있는 수정 사항에 대한 정보는 OpenShift Logging 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
collection:
  logs:
    type: "fluentd" 10
    fluentd: {}

```

- 1 이름은 **instance**이어야 합니다.
- 2 OpenShift Logging 관리 상태입니다. 경우에 따라 OpenShift Logging 기본값을 변경하는 경우 이를 **Unmanaged**로 설정해야 합니다. 그러나 관리되지 않는 배포는 OpenShift Logging이 다시 **Managed** 상태로 될 때까지 업데이트를 받지 않습니다.
- 3 Elasticsearch 구성을 위한 설정입니다. **CR**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 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. **CR**을 사용하여 중복성을 위해 Kibana를 확장하고 Kibana 노드의 CPU 및 메모리를 구성할 수 있습니다. 자세한 내용은 로그 시각화 프로그램 구성을 참조하십시오.
- 10 Fluentd 구성을 위한 설정입니다. **CR**을 사용하여 Fluentd CPU 및 메모리 제한을 구성할 수 있습니다. 자세한 내용은 Fluentd 구성을 참조하십시오.



## 참고

Elasticsearch 컨트롤 플레인 노드(마스터 노드라고도 함)의 최대 수는 3입니다. 3보다 큰 `nodeCount`를 지정하면 OpenShift Container Platform은 마스터, 클라이언트 및 데이터 역할을 가진 마스터 적격 노드인 Elasticsearch 노드 3개를 생성합니다. 추가 Elasticsearch 노드는 클라이언트 및 데이터 역할을 사용하여 데이터 전용 노드로 생성됩니다. 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. 생성을 클릭합니다. 이렇게 하면 OpenShift Logging 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 OpenShift Logging, 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](#)

### 3.2. 설치 후 작업

Kibana를 사용하려면 Kibana에서 데이터를 탐색하고 시각화하기 위해 [Kibana 인덱스 패턴 및 시각화를 수동으로 생성](#)해야 합니다.

클러스터 네트워크 공급자가 네트워크 분리를 적용하는 경우 [OpenShift Logging Operator가 포함된 프로젝트 간에 네트워크 트래픽을 허용](#)합니다.

### 3.3. CLI를 사용하여 OPENSIFT LOGGING 설치

OpenShift Container Platform CLI를 사용하여 OpenShift Elasticsearch Operator 및 Red Hat OpenShift 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 3개의 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.

프로세스

CLI를 사용하여 OpenShift Elasticsearch Operator 및 Red Hat OpenShift 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. Red Hat OpenShift Logging Operator의 네임스페이스를 생성합니다.

a. Red Hat OpenShift Logging Operator를 위한 네임스페이스 오브젝트 YAML 파일(예: olo-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 olo-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 ❶
spec: {}
```

- ❶ openshift-operators-redhat 네임스페이스를 지정해야 합니다.

- b. Operator 그룹 오브젝트를 생성합니다.

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

예를 들면 다음과 같습니다.

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

- c. 서브스크립션 오브젝트 YAML 파일(예: eo-sub.yaml)을 생성하여 네임스페이스에서 OpenShift Elasticsearch Operator를 서브스크립션합니다.

서브스크립션의 예

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: "elasticsearch-operator"
  namespace: "openshift-operators-redhat" ❶
spec:
  channel: "stable-5.1" ❷
  installPlanApproval: "Automatic"
  source: "redhat-operators" ❸
  sourceNamespace: "openshift-marketplace"
  name: "elasticsearch-operator"
```

- ❶ openshift-operators-redhat 네임스페이스를 지정해야 합니다.
- ❷ 5.0, stable 또는 stable-5.<x>를 채널로 지정합니다. 다음 참고 사항을 참조하십시오.
- ❸ redhat-operators를 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.



#### 참고

**stable**을 지정하면 안정적인 최신 릴리스의 현재 버전이 설치됩니다. **installPlanApproval**으로 **stable** 사용: "**automatic**" 는 자동으로 운영 프로그램을 안정적인 최신 주 릴리스 및 마이너 릴리스로 업그레이드합니다.

**stable-5.<x>**를 지정하면 특정 주요 릴리스의 현재 마이너 버전이 설치됩니다. **installPlanApproval**과 함께 **stable-5.<x>** 사용: "**automatic**" 은(는) 자동으로 **x**로 지정한 주요 릴리스 내의 안정적인 최신 마이너 릴리스로 업그레이드합니다.

- 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	VERSION	REPLACES	NAME	PHASE	DISPLAY
default			elasticsearch-operator.5.1.0-202007012112.p0		
OpenShift Elasticsearch Operator	5.1.0-202007012112.p0			Succeeded	
kube-node-lease			elasticsearch-operator.5.1.0-202007012112.p0		
OpenShift Elasticsearch Operator	5.1.0-202007012112.p0			Succeeded	
kube-public			elasticsearch-operator.5.1.0-202007012112.p0		
OpenShift Elasticsearch Operator	5.1.0-202007012112.p0			Succeeded	
kube-system			elasticsearch-operator.5.1.0-202007012112.p0		
OpenShift Elasticsearch Operator	5.1.0-202007012112.p0			Succeeded	
openshift-apiserver-operator			elasticsearch-operator.5.1.0-202007012112.p0		
OpenShift Elasticsearch Operator	5.1.0-202007012112.p0			Succeeded	
openshift-apiserver			elasticsearch-operator.5.1.0-202007012112.p0		
OpenShift Elasticsearch Operator	5.1.0-202007012112.p0			Succeeded	
openshift-authentication-operator			elasticsearch-operator.5.1.0-202007012112.p0		
OpenShift Elasticsearch Operator	5.1.0-202007012112.p0			Succeeded	
openshift-authentication			elasticsearch-operator.5.1.0-202007012112.p0		
OpenShift Elasticsearch Operator	5.1.0-202007012112.p0			Succeeded	
...					

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

4. 다음 오브젝트를 생성하여 Red Hat OpenShift Logging Operator를 설치합니다.
  - a. Red Hat OpenShift Logging Operator를 위한 OperatorGroup 오브젝트 YAML 파일(예: `olo-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 olo-og.yaml
```

c. 서브스크립션 오브젝트 YAML 파일(예: **olo-sub.yaml**)을 생성하여 네임스페이스에서 Red Hat OpenShift Logging Operator를 서브스크립션합니다.

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

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

**2** 5.0, stable 또는 stable-5.<x>를 채널로 지정합니다.

**3** redhat-operators를 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.

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

예를 들면 다음과 같습니다.

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

Red Hat OpenShift Logging Operator는 **openshift-logging** 네임스페이스에 설치됩니다.

d. Operator 설치를 확인합니다.

**openshift-logging** 네임스페이스에 Red Hat OpenShift Logging Operator가 있어야 합니다. 버전 번호가 표시된 것과 다를 수 있습니다.

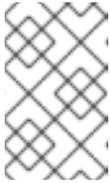
```
$ oc get csv -n openshift-logging
```

## 출력 예

NAMESPACE	NAME	DISPLAY
VERSION	REPLACES	PHASE
...		
openshift-logging	clusterlogging.5.1.0-202007012112.p0	
OpenShift Logging	5.1.0-202007012112.p0	Succeeded
...		

5. OpenShift Logging 인스턴스를 생성합니다.

- Red Hat OpenShift Logging Operator를 위한 인스턴스 오브젝트 YAML 파일(예: `olo-instance.yaml`)을 생성합니다.



## 참고

이 기본 OpenShift Logging 구성은 다양한 환경을 지원해야 합니다. OpenShift Logging 클러스터에 수행할 수 있는 수정 사항에 대한 정보는 OpenShift Logging 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****collection:****logs:****type: "fluentd" 10****fluentd: {}**

- 1 이름은 **instance**이어야 합니다.
- 2 OpenShift Logging 관리 상태입니다. 경우에 따라 OpenShift Logging 기본값을 변경하는 경우 이를 **Unmanaged**로 설정해야 합니다. 그러나 관리되지 않는 배포는 OpenShift Logging이 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. **CR**을 사용하여 중복성을 위해 Kibana를 확장하고 Kibana 노드의 CPU 및 메모리를 구성할 수 있습니다. 자세한 내용은 로그 시각화 프로그램 구성을 참조하십시오.
- 10 Fluentd 구성을 위한 설정입니다. **CR**을 사용하여 Fluentd CPU 및 메모리 제한을 구성할 수 있습니다. 자세한 내용은 Fluentd 구성을 참조하십시오.

## 참고

Elasticsearch 컨트롤 플레인 노드의 최대 수는 3입니다. 3보다 큰 `nodeCount`를 지정하면 OpenShift Container Platform은 마스터, 클라이언트 및 데이터 역할을 가진 마스터 적격 노드인 Elasticsearch 노드 3개를 생성합니다. 추가 Elasticsearch 노드는 클라이언트 및 데이터 역할을 사용하여 데이터 전용 노드로 생성됩니다. 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 olo-instance.yaml
```

이렇게 하면 OpenShift Logging 구성 요소, Elasticsearch 사용자 정의 리소스 및 구성 요소, Kibana 인터페이스가 생성됩니다.

6. openshift-logging 프로젝트에 Pod를 나열하여 설치를 확인합니다.

다음 목록과 유사한 OpenShift Logging, Elasticsearch, Fluentd 및 Kibana에 대한 여러 Pod가 표시됩니다.

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

## 출력 예

```
NAME                                READY STATUS RESTARTS AGE
cluster-logging-operator-66f77fccb-ppzbg 1/1 Running 0 7m
elasticsearch-cdm-ftuhduuw-1-ffc4b9566-q6bhp 2/2 Running 0 2m40s
elasticsearch-cdm-ftuhduuw-2-7b4994dbfc-rd2gc 2/2 Running 0 2m36s
elasticsearch-cdm-ftuhduuw-3-84b5ff7ff8-gqnm2 2/2 Running 0 2m4s
fluentd-587vb                            1/1 Running 0 2m26s
fluentd-7mpb9                             1/1 Running 0 2m30s
```

fluentd-flm6j	1/1	Running	0	2m33s
fluentd-gn4rn	1/1	Running	0	2m26s
fluentd-nlgb6	1/1	Running	0	2m30s
fluentd-snpkt	1/1	Running	0	2m28s
kibana-d6d5668c5-rppqm	2/2	Running	0	2m39s

### 3.4. 설치 후 작업

Kibana를 사용하려면 Kibana에서 데이터를 탐색하고 시각화하기 위해 [Kibana 인덱스 패턴 및 시각화를 수동으로 생성](#)해야 합니다.

클러스터 네트워크 공급자가 네트워크 분리를 적용하는 경우 [OpenShift Logging Operator가 포함된 프로젝트](#) 간에 [네트워크 트래픽을 허용](#)합니다.

#### 3.4.1. 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 로깅을 선택합니다.
2. 관리 → 인덱스 패턴 → 인덱스 패턴 생성을 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.

### 3.4.2. 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 **Red Hat OpenShift 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 정책을 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, openshift-monitoring** 및 **openshift-ingress** 프로젝트에서 **openshift-logging** 프로젝트로 수신할 수 있는 **openshift-logging** 네임스페이스에 네트워크 정책 오브젝트를 만듭니다. 예를 들면 다음과 같습니다.

```

apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-from-openshift-monitoring-ingress-operators-redhat
spec:
  ingress:
  - from:
    - podSelector: {}
  - from:
    - namespaceSelector:
        matchLabels:
          project: "openshift-operators-redhat"
  - from:
    - namespaceSelector:
        matchLabels:
          name: "openshift-monitoring"
  - from:
    - namespaceSelector:
        matchLabels:
          network.openshift.io/policy-group: ingress
  podSelector: {}
  policyTypes:
  - Ingress

```

#### 추가 리소스

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

## 4장. 로깅 배포 구성

### 4.1. 클러스터 로깅 사용자 정의 리소스 정보

OpenShift Logging을 구성하려면 **ClusterLogging** 사용자 정의 리소스(CR)를 사용자 정의합니다.

#### 4.1.1. 클러스터 로깅 사용자 정의 리소스 정보

OpenShift Logging 환경을 변경하려면 **ClusterLogging** 사용자 정의 리소스(CR)를 생성하고 수정합니다.

CR을 작성하거나 수정하기 위한 지침이 이 문서에 적절하게 제공됩니다.

다음은 OpenShift Logging을 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
  
```

```
collection: 6
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 Red Hat OpenShift Logging Operator 관리 상태입니다. **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 제한을 포함한 로그 수집기 설정

## 4.2. 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 Pod**를 특정 노드로 이동할 수 있습니다. **ClusterLogging** 사용자 정의 리소스(CR)의 **spec.collection.log.fluentd** 스탠자를 통해 로그 수집기에 대해 지원되는 모든 수정을 수행할 수 있습니다.

### 4.2.1. 지원되지 않는 구성 정보

지원되는 OpenShift Logging 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 Operator와 Red Hat OpenShift Logging Operator가 차이를 조정하므로 변경한 내용이 사라집니다. Operator는 원래 기본적으로 모든 항목을 정의된 상태로 되돌립니다.



#### 참고

OpenShift Container Platform 설명서에 제시되지 않은 구성이 꼭 필요한 경우 Red Hat OpenShift Logging Operator 또는 OpenShift Elasticsearch Operator를 Unmanaged 상태로 설정해야 합니다. 관리되지 않는 OpenShift Logging 환경은 지원되지 않으며 OpenShift Logging을 Managed 상태로 되돌릴 때까지 업데이트를 받지 않습니다.

### 4.2.2. 로깅 수집기 Pod 보기

Fluentd 로깅 수집기 Pod와 실행 중인 해당 노드를 볼 수 있습니다. Fluentd 로깅 수집기 Pod는 **openshift-logging** 프로젝트에서만 실행됩니다.

## 프로세스

- **openshift-logging** 프로젝트에서 다음 명령을 실행하여 Fluentd 로깅 수집기 Pod 및 세부 정보를 확인합니다.

```
$ 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>
```

## 4.2.3. 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 -n openshift-logging edit ClusterLogging instance
```

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

### 4.2.4. 로그 전달자를 위한 고급 구성

OpenShift Logging에는 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 청크라는 단일 blob에서 로그 데이터를 수집합니다. Fluentd가 청크를 생성할 때 청크는 *스태이*지에 있는 것으로 간주되어 청크가 데이터로 채워집니다. 청크가 가득 차면 Fluentd는 청크를 큐로 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 4.1. 고급 Fluentd 구성 매개변수

매개변수	설명	기본
<b>chunkLimitSize</b>	각 청크의 최대 크기입니다. Fluentd는 이 크기에 도달하면 청크에 데이터 쓰기를 중지합니다. 그런 다음 Fluentd는 청크를 큐로 보내고 새 청크를 엽니다.	<b>8m</b>

매개변수	설명	기본
<b>totalLimitSize</b>	스테이지와 큐의 총 크기인 버퍼의 최대 크기입니다. 버퍼 크기가 이 값을 초과하면 Fluentd는 청크로의 데이터 추가를 중지하고 오류와 함께 실패합니다. 청크에 없는 모든 데이터는 손실됩니다.	<b>8G</b>
<b>flushInterval</b>	청크 플러시 간격입니다. <b>s</b> (초), <b>m</b> (분), <b>h</b> (시간) 또는 <b>d</b> (일)를 사용할 수 있습니다.	<b>1s</b>
<b>flushMode</b>	플러시를 수행하는 방법: <ul style="list-style-type: none"> <li>● <b>lazy: timekey</b> 매개 변수를 기반으로 청크를 플러시합니다. <b>timekey</b> 매개 변수는 수정할 수 없습니다.</li> <li>● <b>interval: flushInterval</b> 매개 변수를 기반으로 청크를 플러시합니다.</li> <li>● <b>immediate</b>: 데이터가 청크에 추가된 직후 청크를 플러시합니다.</li> </ul>	간격
<b>flushThreadCount</b>	청크 플러시를 수행하는 스레드 수입니다. 스레드 수를 늘리면 플러시 처리량이 향상되어 네트워크 대기 시간이 숨겨집니다.	<b>2</b>
<b>overflowAction</b>	큐가 가득 찼을 때 청크 동작: <ul style="list-style-type: none"> <li>● <b>throw_exception</b>: 로그에 표시할 예외를 높입니다.</li> <li>● <b>block</b>: 전체 버퍼 문제가 해결될 때까지 데이터 청크를 중지합니다.</li> <li>● <b>drop_oldest_chunk</b>: 가장 오래된 청크를 삭제하여 새로 들어오는 청크를 수락합니다. 오래된 청크는 새로운 청크보다 가치가 적습니다.</li> </ul>	블록
<b>retryMaxInterval</b>	<b>exponential_backoff</b> 재시도 방법의 최대 시간(초)입니다.	<b>300s</b>

매개변수	설명	기본
<b>retryType</b>	플러시 실패 시 재시도 방법: <ul style="list-style-type: none"> <li>● <b>exponential_backoff</b>: 플러시 재시도 간격을 늘립니다. Fluentd는 <b>retry_max_interval</b> 매개변수에 도달할 때까지 다음 재시도까지 대기하는 시간을 두 배로 늘립니다.</li> <li>● <b>periodic: retryWait</b> 매개변수를 기반으로 플러시를 주기적으로 재시도 합니다.</li> </ul>	<b>exponential_backoff</b>
<b>retryWait</b>	다음 청크 플러시 전의 시간(초)입니다.	<b>1s</b>

Fluentd 청크 수명 주기에 대한 자세한 내용은 Fluentd 문서의 [버퍼 플러그인](#)을 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. 체크 플러시 간격을 지정합니다.
3. `lazy`, `interval` 또는 `immediate` 등 체크 플러시를 수행할 방법을 지정합니다.
4. 체크 플러시에 사용할 스레드 수를 지정합니다.
5. `throw_exception`, `block` 또는 `drop_oldest_chunk` 등 큐가 가득 찼을 때의 체크 동작을 지정합니다.
6. `exponential_backoff` 체크 플러시 방법의 최대 간격(초)을 지정합니다.
7. 체크 플러시 실패 시 재시도 유형을 `exponential_backoff` 또는 `periodic`으로 지정합니다.
8. 다음 체크 플러시 전 시간(초)을 지정합니다.
9. 체크 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>
```

#### 4.2.5. 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** 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 CR**은 로그 데이터를 내부 Elasticsearch 클러스터로 전달하고 **ClusterLogging CR**에서 **logStore** 구성 요소를 제거합니다. 이 경우 로그 데이터를 저장할 내부 Elasticsearch 클러스터가 표시되지 않습니다. 이 경우 데이터 손실이 발생할 수 있습니다.

## 프로세스

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

```
$ oc edit ClusterLogging instance
```

- ClusterLogging CR**에서 **logStore**, **visualization** 스탠자를 제거하십시오.
- ClusterLogging CR**의 **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: {}
```

- Fluentd Pod가 재배포되었는지 확인합니다.

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

## 추가 리소스

- [타사 시스템에 로그 전달](#)

## 4.3. 로그 저장소 구성

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

다음은 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 최소 16G의 메모리가 필요합니다. 초기 OpenShift Container Platform 노드 세트는 Elasticsearch 클러스터를 지원하기에 충분히 크지 않을 수 있습니다. 권장 메모리 이상에서 각 Elasticsearch 노드에 대해 최대 64G까지 실행하려면 OpenShift Container Platform 클러스터에 노드를 추가해야 합니다.

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

### 4.3.1. 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 Logging은 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 Forwarding API를 사용하여 로그 전달을 참조하십시오](#).

### 4.3.2. 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** 매개 변수를 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 CR의 **rollover.maxAge** 값보다 오래되었습니다.
- 인덱스 크기가 40GB × 기본 shard 수보다 큼니다.
- 인덱스 문서 수가 40960KB × 기본 shard 수보다 큼니다.

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

#### 사전 요구 사항

- OpenShift Logging 및 OpenShift Elasticsearch Operator가 설치되어 있어야 합니다.

#### 절차

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 1d입니다. **maxAge**보다 오래된
2. **Elasticsearch** 사용자 정의 리소스(CR)에서 설정을 확인할 수 있습니다.  
 예를 들어 Red Hat OpenShift Logging Operator가 8시간마다 인프라 로그의 활성 인덱스를 롤오버하는 설정이 포함된 보존 정책을 구성하기 위해 다음 **Elasticsearch CR**을 업데이트했고, 롤오버된 인덱스는 롤오버 후 7일이 지나면 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 CR**에서 설정한 **maxAge**입니다.
- 3 인덱스를 롤오버할 때 고려해야 할 OpenShift Container Platform의 인덱스 수명입니다. 이 값은 **ClusterLogging CR**에서 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 CR**에서 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
------	----------	---------	--------	---------------	-----

elasticsearch-im-app	*15 *****	False	0	<none>	4s
elasticsearch-im-audit	*15 *****	False	0	<none>	4s
elasticsearch-im-infra	*15 *****	False	0	<none>	4s

### 4.3.3. 로그 저장소에 대한 CPU 및 메모리 요청 구성

각 구성 요소 사양을 통해 CPU 및 메모리 요청을 조정할 수 있습니다. OpenShift 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

- OpenShift Logging 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: ①
    resources:
      limits: ②
        memory: "32Gi"
      requests: ③
        cpu: "1"
        memory: "16Gi"
    proxy: ④
    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 16Gi이고 CPU 요청 시 1입니다.
- 2 포드에서 사용할 수 있는 최대 리소스 양입니다.
- 3 Pod를 예약하는 데 필요한 최소 리소스입니다.
- 4 필요에 따라 Elasticsearch 프록시에 대한 CPU 및 메모리 요청을 지정합니다. 이러한 값을 비워 두면 OpenShift Elasticsearch Operator는 대부분의 배포에 충분한 기본값을 설정합니다. 기본값은 메모리 요청 시 256Mi이고 CPU 요청 시 100m입니다.

Elasticsearch 메모리 양을 조정할 때 요청 및 제한 모두에 동일한 값을 사용해야 합니다.

예를 들면 다음과 같습니다.

```
resources:
  limits: 1
    memory: "32Gi"
  requests: 2
    cpu: "8"
    memory: "32Gi"
```

- 1 리소스의 최대 양입니다.
- 2 필요한 최소량.

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.

#### 4.3.4. 로그 저장소에 대한 복제 정책 구성

Elasticsearch shard가 클러스터의 데이터 노드에 복제되는 방법을 정의할 수 있습니다.

사전 요구 사항

- OpenShift Logging 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.

### 4.3.5. 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.

### 4.3.6. 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

- OpenShift Logging 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: block에 설명된 원시 블록 볼륨을 사용하지 마십시오. Elasticsearch는 원시 블록 볼륨을 사용할 수 없습니다.

### 4.3.7. 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

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

1. emptyDir을 지정하려면 ClusterLogging CR을 편집합니다.

```

spec:
  logStore:
    type: "elasticsearch"

```

```

elasticsearch:
  nodeCount: 3
  storage: {}

```

### 4.3.8. 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

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

클러스터를 롤링 재시작하려면 다음을 수행합니다.

1. **openshift-logging** 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 플리시를 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 벨런싱을 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"}}}}}'

```

#### 4.3.9. 로그 저장소 서비스를 경로로 노출

기본적으로 OpenShift Logging과 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

- OpenShift Logging 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 제암호화 경로에 필요한 `spec.tls.key`, `spec.tls.certificate` 및 `spec.tls.caCertificate` 매개 변수를 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>"
}
```

#### 4.4. 로그 시각화 프로그램 구성

OpenShift Container Platform은 Kibana를 사용하여 OpenShift Logging으로 수집된 로그 데이터를 표시합니다.

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

#### 4.4.1. CPU 및 메모리 제한 구성

OpenShift Logging 구성 요소를 사용하면 CPU 및 메모리 제한을 모두 조정할 수 있습니다.

프로세스

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

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

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: openshift-logging
...
spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      resources: ①
        limits:
          memory: 16Gi
        requests:
          cpu: 200m
          memory: 16Gi
      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
      collection:
        logs:
          type: "fluentd"
          fluentd:
```

```
resources: 4
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 수집기에 대한 CPU 및 메모리 제한 및 요청을 지정합니다.

#### 4.4.2. 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.

### 4.5. OPENSIFT LOGGING 스토리지 구성

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

#### 4.5.1. OpenShift Logging 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.

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.

### 4.5.2. 추가 리소스

- [로그 저장소에 대한 영구 스토리지 구성](#)

## 4.6. OPENSIFT LOGGING 구성 요소에 대한 CPU 및 메모리 제한 구성

필요에 따라 각 OpenShift Logging 구성 요소에 대한 CPU 및 메모리 제한을 모두 구성할 수 있습니다.

### 4.6.1. CPU 및 메모리 제한 구성

OpenShift Logging 구성 요소를 사용하면 CPU 및 메모리 제한을 모두 조정할 수 있습니다.

#### 프로세스

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

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

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

```

spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      resources: ①
      limits:
        memory: 16Gi
      requests:
        cpu: 200m
        memory: 16Gi
    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
  collection:
    logs:
      type: "fluentd"
      fluentd:
        resources: ④
        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 수집기에 대한 CPU 및 메모리 제한 및 요청을 지정합니다.

## 4.7. 허용 오차를 사용하여 OPENSIFT LOGGING POD 배치 제어

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

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

**key**는 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 OpenShift Logging CR

```

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

...

spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      tolerations: 1
      - key: "logging"
        operator: "Exists"
        effect: "NoExecute"
        tolerationSeconds: 6000
    resources:
      limits:
        memory: 16Gi
      requests:
        cpu: 200m
        memory: 16Gi
      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.

#### 4.7.1. 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 pair입니다. 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

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

1. 다음 명령을 사용하여 OpenShift Logging 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 NoExecute로 node1에 taint를 배치합니다. NoExecute 효과가 있는 노드는 taint와 일치하는 Pod만 스케줄링하고 일치하지 않는 기존 Pod는 제거합니다.

2. Elasticsearch Pod에 대한 허용 오차를 구성하려면 **ClusterLogging CR**의 **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 Operator**를 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.

#### 4.7.2. 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

- OpenShift Logging 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 Operator**를 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 **taint**와 일치합니다. 이 허용 오차가 있는 Pod는 **node1**에 스케줄링할 수 있습니다.

### 4.7.3. 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 **taint**를 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

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

#### 프로세스

1. 다음 명령을 사용하여 로깅 수집기 Pod에서 로깅 수집기 Pod를 스케줄링할 노드에 **taint**를 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** 스태ن자를 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 Operator**를 지정합니다.
- 3 **NoExecute** 효과를 지정합니다.
- 4 선택적으로 **tolerationSeconds** 매개변수를 지정하여 Pod가 제거되기 전까지 노드에 바인딩 되는 시간을 설정합니다.

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

#### 4.7.4. 추가 리소스

- [노드 테인트를 사용하여 Pod 배치 제어](#)

## 4.8. 노드 선택기로 OPENSIFT LOGGING 리소스 이동

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

### 4.8.1. OpenShift Logging 리소스 이동

Elasticsearch 및 Kibana와 같은 OpenShift Logging 구성 요소 용 Pod를 다른 노드에 배포하도록 OpenShift 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

- OpenShift Logging 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
    logStore:
      elasticsearch:
        nodeCount: 3
        nodeSelector: 1
        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: 2
        node-role.kubernetes.io/infra: "
        proxy:
          resources: null
        replicas: 1
        resources: null
        type: kibana
...

```

1 2 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 v1.20.0
ip-10-0-139-146.us-east-2.compute.internal Ready master    60m v1.20.0
ip-10-0-139-192.us-east-2.compute.internal Ready worker    51m v1.20.0
ip-10-0-139-241.us-east-2.compute.internal Ready worker    51m v1.20.0
ip-10-0-147-79.us-east-2.compute.internal Ready worker    51m v1.20.0
ip-10-0-152-241.us-east-2.compute.internal Ready master    60m v1.20.0
ip-10-0-139-48.us-east-2.compute.internal Ready infra     51m v1.20.0
```

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-nkx17	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
kibana-7d85dcffc8-bfpfp	2/2	Running	0	43s	10.131.0.22	ip-10-0-139-48.us-east-2.compute.internal
		NOMINATED NODE READINESS GATES				
		<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-nkx17	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

## 4.9. 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.

### 4.9.1. OpenShift Logging을 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. OpenShift Logging이 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 0을 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. **systemd**는 디렉토리가 없는 경우 디렉토리를 생성합니다.
    - 디렉토리가 존재하는 경우 **/var/log/journal/**에 로그를 저장하기 위한 **auto**입니다. 존재하지 않는 경우 **systemd**는 **/run/systemd/journal**에 로그를 임시 저장합니다.
    - 로그를 저장하지 않는 **none**입니다. **systemd**는 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 **systemd**가 즉시 동기화됩니다. 기본값은 **1s**입니다.
  7. 저널이 사용할 수 있는 최대 크기를 지정합니다. 기본값은 **8G**입니다.
  8. 시스템에서 사용 가능한 디스크 공간을 지정합니다. 기본값은 **20%**입니다.
  9. **/var/log/journal**에 지속적으로 저장된 개별 저널 파일의 최대 크기를 지정합니다. 기본값은 **10M**입니다.



**참고**

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

시스템 설정에 대한 자세한 내용은

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

2. **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 ❶
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker ❷
  name: 40-worker-custom-journald ❸
spec:
  config:
    ignition:
      config: {}
      security:
        tls: {}
        timeouts: {}
      version: 3.2.0
    networkd: {}
    passwd: {}
    storage:
      files:
        - contents:
            source: data:text/plain;charset=utf-8;base64,${jrnl_cnf} ❹
            verification: {}
          filesystem: root
          mode: 0644 ❺
          path: /etc/systemd/journald.conf.d/custom.conf
        osImageURL: ""
EOF
```

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

4. 다음 명령을 실행하여 머신 구성을 생성합니다.

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

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

5. 다음 명령을 실행하여 각 노드에 새로 렌더링된 구성의 롤아웃 상태를 모니터링합니다.

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

- 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

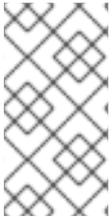
...

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

## 4.10. 유지보수 및 지원

### 4.10.1. 지원되지 않는 구성 정보

지원되는 OpenShift Logging 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 Operator와 Red Hat OpenShift Logging Operator가 차이를 조정하므로 변경한 내용이 사라집니다. Operator는 원래 기본적으로 모든 항목을 정의된 상태로 되돌립니다.



#### 참고

OpenShift Container Platform 설명서에 제시되지 않은 구성이 꼭 필요한 경우 Red Hat OpenShift Logging Operator 또는 OpenShift Elasticsearch Operator를 Unmanaged 상태로 설정해야 합니다. 관리되지 않는 OpenShift Logging 환경은 지원되지 않으며 OpenShift Logging을 Managed 상태로 되돌릴 때까지 업데이트를 받지 않습니다.

### 4.10.2. 지원되지 않는 로깅 구성

다음 구성 요소를 수정하려면 Red Hat OpenShift Logging Operator를 관리되지 않음 상태로 설정해야 합니다.

- Elasticsearch CR
- Kibana 배포
- fluent.conf 파일
- Fluentd 데몬 세트

다음 구성 요소를 수정하려면 OpenShift Elasticsearch Operator를 관리되지 않음 상태로 설정해야 합니다.

- Elasticsearch 배포 파일.

명시적으로 지원되지 않는 경우는 다음과 같습니다.

- 기본 로그 회전 구성. 기본 로그 회전을 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.
- 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.

### 4.10.3. 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, Red Hat OpenShift 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 재정의의 목록을 제공할 수 있습니다. 구성 요소에 대해 `spec.overrides[].unmanaged` 매개변수를 `true`로 설정하면 클러스터 업그레이드가 차단되고 CVO 재정의가 설정된 후 관리자에게 경고합니다.

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



### 주의

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

## 5장. 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 Logging](#)을 설치하는 것이 좋습니다. OpenShift Logging은 노드 시스템 감사 로그, 애플리케이션 컨테이너 로그 및 인프라 로그와 같은 OpenShift Container Platform 클러스터의 모든 로그를 전용 로그 저장소로 집계할 수 있습니다. 그런 다음 [Kibana 인터페이스](#)를 통해 로그 데이터를 쿼리, 검색 및 시각화할 수 있습니다. 리소스 로그는 OpenShift Logging 로그 저장소에 액세스하지 않습니다.

### 5.1. 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 → Pod로 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. 로그를 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.

## 6장. KIBANA를 사용하여 클러스터 로그 보기

OpenShift Logging에는 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 문서](#)를 참조하십시오.



### 참고

감사 로그는 기본적으로 내부 OpenShift Container Platform Elasticsearch 인스턴스에 저장되지 않습니다. Kibana에서 감사 로그를 보려면 [Log Forwarding API](#)를 사용하여 감사 로그에 **default** 출력을 사용하는 파이프라인을 구성해야 합니다.

### 6.1. 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 API](#)를 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 로깅을 선택합니다.
2. 관리 → 인덱스 패턴 → 인덱스 패턴 생성을 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.

## 6.2. KIBANA에서 클러스터 로그 보기

Kibana 웹 콘솔에서 클러스터 로그를 봅니다. 이 문서의 범위를 벗어난 Kibana에서 데이터를 보고 시각화하는 방법입니다. 자세한 내용은 [Kibana 설명서](#)를 참조하십시오.

### 사전 요구 사항

- OpenShift Logging 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 로깅을 선택합니다.
2. OpenShift Container Platform 콘솔에 로그인할 때 사용하는 것과 동일한 자격 증명을 사용하여 로그인합니다.  
Kibana 인터페이스가 시작됩니다.

3. Kibana에서 검색을 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 6.1. Kibana의 샘플 인프라 로그 항목

```
{
  "_index": "infra-000001",
  "_type": "_doc",
  "_id": "YmJmYTBINDkZTRmLTliMGQtMjE3NmFiOGUyOWM3",
  "_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:v4.7",
      "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": "YmJmYTBINDktMDMGQtMjE3NmFiOGUyOWM3",
    "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  
]  
}
```

## 7장. 타사 시스템에 로그 전달

기본적으로 OpenShift Container Platform 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 Cluster Log Forwarder를 구성할 필요가 없습니다.

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

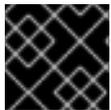


### 참고

내부 로그 저장소로 감사 로그를 보내려면 [로그 저장소에 감사 로그 전달](#)의 설명에 따라 Cluster Log Forwarder를 사용하십시오.

로그를 외부로 전달할 때 Red Hat OpenShift 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 프로토콜**을 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 Cluster Log Forwarder를 사용할 수 없습니다.

## 7.1. 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 출력과 **파이프라인**의 조합을 사용해야 합니다. 입력을 사용하여 특정 프로젝트와 관련된 애플리케이션 로그를 끝점으로 전달할 수도 있습니다.

- 출력은 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 Red Hat OpenShift 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.** **auditd**에서 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** CR 오브젝트가 있는 경우 **default** 출력이 있는 파이프라인이 없으면 로그가 기본 **Elasticsearch** 인스턴스로 전달되지 않습니다.
- 기본적으로 OpenShift Container Platform 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 API**를 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 Logging**은 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
      parse: json ⑧
      labels:
        secure: "true" ⑨
        datacenter: "east"
    - name: infrastructure-logs ⑩
      inputRefs:
        - infrastructure
      outputRefs:
        - elasticsearch-insecure
      labels:
        datacenter: "west"
    - name: my-app ⑪
      inputRefs:
        - my-app-logs
      outputRefs:
        - default
    - inputRefs: ⑫
      - 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: 구조화된 필드의 JSON 오브젝트로 구조화된 JSON 로그 항목을 전달합니다. 로그 항목에 유효한 구조화된 JSON이 포함되어야 합니다. 그렇지 않으면 OpenShift Logging이 **structured** 필드를 제거하고 대신 기본 인덱스인 **app-00000x**로 로그 항목을 보냅니다.
- 9 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다. "true"와 같은 인용 값은 부울 값이 아닌 문자열 값으로 인식됩니다.
- 10 인프라 로그를 안전하지 않은 외부 Elasticsearch 인스턴스로 전송하는 파이프라인 구성:
- 11 **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 default입니다.
- 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

**12** 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 **Fluentd**는 계속 로그를 수집하여 버퍼에 저장합니다. 로그 집계기를 사용할 수 있게 되면 버퍼링된 로그를 포함하여 로그 전달이 재개됩니다. 버퍼가 완전히 채워지면 **Fluentd**는 로그 수집을 중지합니다. **OpenShift Container Platform**은 로그를 회전시켜 삭제합니다. **Fluentd** 데몬 세트 또는 **pod**에 버퍼 크기를 조정하거나 **PVC**(영구 볼륨 클레임)를 추가할 수 없습니다.

## 7.2. 지원되는 로그 데이터 출력 유형

**Red Hat OpenShift Logging 5.0**은 로그 데이터를 대상 로그 수집기로 전송하는 데 필요한 다음과 같은 출력 유형 및 프로토콜을 제공합니다.

**Red Hat**은 다음 표에 표시된 각 조합을 테스트합니다. 그러나 이러한 프로토콜을 수집하는 광범위한 범위 대상 로그 수집기로 로그 데이터를 전송할 수 있어야 합니다.

출력 대상	프로토콜	테스트에 사용
fluentdForward	fluentd forward v1	fluentd 1.7.4 logstash 7.10.1
elasticsearch	elasticsearch	Elasticsearch 6.8.1 Elasticsearch 7.10.1
syslog	RFC-3164, RFC-5424	rsyslog 8.37.0-9.el7
kafka	kafka 0.11	kafka 2.4.1



### 참고

이전에는 **syslog** 출력이 **RFC-3164**만 지원했습니다. 현재 **syslog** 출력에는 **RFC-5424**에 대한 지원이 추가되었습니다.

## 7.3. 외부 ELASTICSEARCH 인스턴스로 로그 전달

선택적으로 내부 **OpenShift Container Platform Elasticsearch** 인스턴스에 추가하거나 대신 외부 **Elasticsearch** 인스턴스에 로그를 전달할 수 있습니다. **OpenShift Container Platform**에서 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 Red Hat OpenShift Logging Operator용으로 예약되므로 오류 메시지가 나타납니다.



참고

내부 OpenShift Container Platform Elasticsearch 인스턴스에만 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
      parse: json 11
      labels:
        myLabel: "myValue" 12
    - name: infrastructure-audit-logs 13
      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: 구조화된 필드의 JSON 오브젝트로 구조화된 JSON 로그 항목을 전달합니다. 로그 항목에 유효한 구조화된 JSON이 포함되어야 합니다. 그렇지 않으면 OpenShift Logging이 **structured** 필드를 제거하고 대신 기본 인덱스인 **app-00000x**로 로그 항목을 보냅니다.
- 12 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.
- 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
```

Red Hat OpenShift Logging Operator는 Fluentd Pod를 재배포합니다. Pod가 재배포되지 않으면 Fluentd Pod를 삭제하여 강제로 재배포할 수 있습니다.

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

## 7.4. FLUENTD 정방향 프로토콜을 사용하여 로그 전달

Fluentd 전달 프로토콜을 사용하여 기본 Elasticsearch 로그 저장소 대신 또는 기본 Elasticsearch 로그 저장소에 더하여 해당 프로토콜을 수락하도록 구성된 외부 로그 집계기로 로그 사본을 보낼 수 있습니다. OpenShift Container Platform에서 로그를 수신하도록 외부 로그 집계기를 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
      parse: json 11
      labels:
        clusterId: "C1234" 12
    - name: forward-to-fluentd-insecure 13
      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: 구조화된 필드의 JSON 오브젝트로 구조화된 JSON 로그 항목을 전달합니다. 로그 항목에 유효한 구조화된 JSON이 포함되어야 합니다. 그렇지 않으면 OpenShift Logging이 **structured** 필드를 제거하고 대신 기본 인텍스인 **app-00000x**로 로그 항목을 보냅니다.
- 12 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.
- 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
```

Red Hat OpenShift Logging Operator는 Fluentd Pod를 재배포합니다. Pod가 재배포되지 않으면 Fluentd Pod를 삭제하여 강제로 재배포할 수 있습니다.

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

## 7.5. 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

- 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
        proclD: 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
  parse: json 12
  labels:
    secure: "true" 13
    syslog: "east"
- name: syslog-west 14
  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: 구조화된 필드의 JSON 오브젝트로 구조화된 JSON 로그 항목을 전달합니다. 로그 항목에 유효한 구조화된 JSON이 포함되어야 합니다. 그렇지 않으면 OpenShift Logging이 **structured** 필드를 제거하고 대신 기본 인덱스인 **app-00000x**로 로그 항목을 보냅니다.
- 13 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다. "true"와 같은 인용 값은 부울 값이 아닌 문자열 값으로 인식됩니다.

**14** 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
```

Red Hat OpenShift Logging Operator는 Fluentd Pod를 재배포합니다. Pod가 재배포되지 않으면 Fluentd Pod를 삭제하여 강제로 재배포할 수 있습니다.

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

### 7.5.1. Syslog 매개변수

**syslog** 출력에 대해 다음을 구성할 수 있습니다. 자세한 내용은 [syslog RFC3164](#) 또는 [RFC5424 RFC](#)를 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 서버 시스템의 경우 **7** 또는 **news**
  - UUCP 하위 시스템의 경우 **8** 또는 **uucp**
  - 시계 데몬의 경우 **9** 또는 **cron**
  - 보안 인증 메시지의 경우 **10** 또는 **authpriv**
  - FTP 데몬의 경우 **11** 또는 **ftp**
  - NTP 하위 시스템의 경우 **12** 또는 **ntp**
  - syslog 감사 로그의 경우 **13** 또는 **security**

- syslog 경고 로그의 경우 **14** 또는 **console**
- 스케줄링 데몬의 경우 **15** 또는 **solaris-cron**
- 로컬에서 사용되는 시설의 경우 **16 - 23** 또는 **local0 - local7**
- **Optional. payloadKey:** syslog 메시지의 페이로드로 사용할 레코드 필드입니다.



#### 참고

**payloadKey** 매개변수를 구성하면 다른 매개 변수가 syslog로 전달되지 않습니다.

- **rfc:** syslog를 사용하여 로그를 전송하는 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.

### 7.5.2. 추가 RFC5424 syslog 매개 변수

RFC5424에는 다음 매개 변수가 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.
- **proclD:** PROCID는 자유 텍스트 문자열입니다. 값이 변경되면 syslog 보고가 중단되었음을 나타냅니다. **RFC5424**에 대해 지정해야 합니다.

## 7.6. 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
      parse: json 11
      labels:
        logType: "application" 12
    - name: infra-topic 13
      inputRefs:
      - infrastructure
      outputRefs:
      - infra-logs
      labels:
        logType: "infra"
    - name: audit-topic
      inputRefs:
      - audit
      outputRefs:
      - audit-logs
      - default 14
      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: 구조화된 필드의 JSON 오브젝트로 구조화된 JSON 로그 항목을 전달합니다. 로그 항목에 유효한 구조화된 JSON이 포함되어야 합니다. 그렇지 않으면 OpenShift Logging이 **structured** 필드를 제거하고 대신 기본 인덱스인 **app-00000x**로 로그 항목을 보냅니다.
  - 12 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.
  - 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.
  - 14 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
```

Red Hat OpenShift Logging Operator는 Fluentd Pod를 재배포합니다. Pod가 재배포되지 않으면 Fluentd Pod를 삭제하여 강제로 재배포할 수 있습니다.

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

### 7.7. 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
  parse: json 11
  labels:
    project: "my-project" 12
- name: forward-to-fluentd-secure 13
  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: 구조화된 필드의 JSON 오브젝트로 구조화된 JSON 로그 항목을 전달합니다. 로그 항목에 유효한 구조화된 JSON이 포함되어야 합니다. 그렇지 않으면 OpenShift Logging이 **structured** 필드를 제거하고 대신 기본 인덱스인 **app-00000x**로 로그 항목을 보냅니다.

- 12 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.
- 13 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
```

## 7.8. 특정 POD에서 애플리케이션 로그 전달

클러스터 관리자는 Kubernetes Pod 레이블을 사용하여 특정 Pod에서 로그 데이터를 수집하여 로그 수집기로 전달할 수 있습니다.

다양한 네임스페이스의 다른 pod와 함께 실행되는 pod로 구성된 애플리케이션이 있다고 가정합니다. 이러한 pod에 애플리케이션을 식별하는 레이블이 있는 경우 로그 데이터를 특정 로그 수집기로 수집하고 출력할 수 있습니다.

Pod 라벨을 지정하려면 하나 이상의 **matchLabels** 키-값 쌍을 사용합니다. 여러 키-값 쌍을 지정하는 경우 Pod를 모두 선택할 수 있어야 합니다.

### 프로세스

1. **ClusterLogForwarder** CR(사용자 정의 리소스) YAML 파일을 생성합니다.
2. YAML 파일에서 다음 예와 같이 **inputs[].name.application.selector.matchLabels**에서 간단한 동일성 기반 선택기를 사용하여 Pod 레이블을 지정합니다.

### ClusterLogForwarder CR YAML 파일의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: instance 1
  namespace: openshift-logging 2
spec:
  pipelines:
    - inputRefs: [ myAppLogData ] 3
      outputRefs: [ default ] 4
      parse: json 5
  inputs: 6
    - name: myAppLogData
```

```

application:
  selector:
    matchLabels: 7
    environment: production
    app: nginx
  namespaces: 8
  - app1
  - app2
outputs: 9
  - default
  ...

```

- 1 ClusterLogForwarder CR의 이름은 **instance**여야 합니다.
  - 2 ClusterLogForwarder CR의 네임스페이스는 **openshift-logging**이어야 합니다.
  - 3 **inputs[].name**에서 하나 이상의 쉼표로 구분된 값을 지정합니다.
  - 4 **outputs[]**에서 하나 이상의 쉼표로 구분된 값을 지정합니다.
  - 5 선택 사항: 구조화된 필드의 JSON 오브젝트로 구조화된 JSON 로그 항목을 전달합니다. 로그 항목에 유효한 구조화된 JSON이 포함되어야 합니다. 그렇지 않으면 OpenShift Logging이 **structured** 필드를 제거하고 대신 기본 인덱스인 **app-00000x**로 로그 항목을 보냅니다.
  - 6 고유한 pod 레이블 집합이 있는 각 애플리케이션에 대해 고유한 **inputs[].name**을 정의합니다.
  - 7 수집하려는 로그 데이터가 있는 Pod 라벨의 키-값 쌍을 지정합니다. 키뿐만 아니라 키와 값 모두를 지정해야 합니다. 선택하려면 Pod가 모든 키-값 쌍과 일치해야 합니다.
  - 8 선택 사항: 하나 이상의 네임스페이스를 지정합니다.
  - 9 로그 데이터를 전달할 출력을 하나 이상 지정합니다. 여기에 표시된 **default** 출력 (선택 사항)은 로그 데이터를 내부 Elasticsearch 인스턴스로 전송합니다.
3. 선택 사항: 로그 데이터 수집을 특정 네임스페이스로 제한하려면 위 예제와 같이 **inputs[].name.application.namespaces** 를 사용합니다.
  4. 선택 사항: 다른 Pod 라벨이 있는 추가 애플리케이션에서 동일한 파이프라인으로 로그 데이터를 보낼 수 있습니다.
    - a. Pod 레이블의 고유한 조합마다 표시된 항목과 유사한 추가 **inputs[].name** 섹션을 생성합니다.
    - b. 이 애플리케이션의 Pod 레이블과 일치하도록 **selectors**를 업데이트합니다.
    - c. 새 **inputs[].name** 값을 **inputRefs**에 추가합니다. 예를 들면 다음과 같습니다.
 

```

          - inputRefs: [ myAppLogData, myOtherAppLogData ]
          
```
  5. CR 오브젝트를 생성합니다.

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

- Kubernetes의 `matchLabels`에 대한 자세한 내용은 [세트 기반 요구 사항을 지원하는 리소스를 참조](#) 하십시오.

## 7.9. 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 매개변수**를 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 forward는 이러한 서버 노드를 라운드 로빈 순서로 사용합니다.

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

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

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

Red Hat OpenShift Logging Operator는 Fluentd Pod를 재배포합니다. Pod가 재배포되지 않으면 Fluentd Pod를 삭제하여 강제로 재배포할 수 있습니다.

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

## 7.10. 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 **체크로** 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 **syslogRFC3164**를 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 전송하는 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>** 스텐자에서 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
```

Red Hat OpenShift Logging Operator는 Fluentd Pod를 재배포합니다. Pod가 재배포되지 않으면 Fluentd Pod를 삭제하여 강제로 재배포할 수 있습니다.

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

## 8장. JSON 로깅 활성화

JSON 문자열을 구조화된 오브젝트로 구문 분석하도록 Log Forwarding API를 구성할 수 있습니다.

### 8.1. JSON 로그 구문 분석

JSON 로그를 포함한 로그는 일반적으로 **message** 필드 내에 문자열로 표시됩니다. 따라서 사용자는 JSON 문서 내의 특정 필드를 쿼리하기가 어렵습니다. OpenShift Logging의 Log Forwarding API를 사용하면 JSON 로그를 구조화된 오브젝트로 구문 분석하고 OpenShift Logging 관리 Elasticsearch 또는 Log Forwarding API에서 지원하는 기타 타사 시스템으로 전달할 수 있습니다.

이 작동 방식을 설명하려면 다음과 같이 구조화된 JSON 로그 항목이 있다고 가정합니다.

구조화된 JSON 로그 항목 예

```
{"level":"info","name":"fred","home":"bedrock"}
```

일반적으로 **ClusterLogForwarder** CR(사용자 정의 리소스)은 해당 로그 항목을 **message** 필드에 전달합니다. **message** 필드에는 다음 예와 같이 JSON 로그 항목과 동등한 JSON 인용 문자열이 포함되어 있습니다.

**message** 필드 예

```
{"message": "{\"level\":\"info\",\"name\":\"fred\",\"home\":\"bedrock\"\",  
  \"more fields...\"}
```

JSON 로그를 구문 분석할 수 있도록 다음 예제와 같이 **parse: json**을 **ClusterLogForwarder** CR의 파이프라인에 추가합니다.

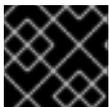
**parse: json**을 보여주는 코드 조각 예

```
pipelines:  
- inputRefs: [ application ]  
  outputRefs: myFluentd  
  parse: json
```

**parse: json**을 사용하여 JSON 로그를 구문 분석하는 경우 CR은 다음 예와 같이 **structured** 필드에 JSON 구조 로그 항목을 복사합니다. 원본 **message** 필드를 수정하지 않습니다.

구조화된 JSON 로그 항목이 포함된 **structured** 출력 예

```
{"structured": { "level": "info", "name": "fred", "home": "bedrock" },  
  "more fields..."}
```



#### 중요

로그 항목에 유효한 구조화된 JSON이 포함되어 있지 않으면 **structured** 필드가 없습니다.

특정 로깅 플랫폼에 대한 JSON 로그를 구문 분석할 수 있도록 하려면 [타사 시스템으로 로그 전달](#)을 참조하십시오.

## 8.2. ELASTICSEARCH의 JSON 로그 데이터 구성

JSON 로그가 두 개 이상의 스키마를 따르는 경우 단일 인덱스에 저장하면 유형 충돌 및 카디널리티 문제가 발생할 수 있습니다. 이를 방지하려면 **ClusterLogForwarder** 사용자 정의 리소스(CR)를 구성하여 각 스키마를 단일 출력 정의로 그룹화해야 합니다. 이렇게 하면 각 스키마가 별도의 인덱스로 전달됩니다.



### 중요

OpenShift Logging에서 관리하는 기본 Elasticsearch 인스턴스로 JSON 로그를 전달하면 구성에 따라 새 인덱스가 생성됩니다. 너무 많은 인덱스를 보유하는 것과 관련된 성능 문제를 방지하려면 공통 스키마로 표준화하여 가능한 스키마 수를 유지하는 것이 좋습니다.

### 구조 유형

**ClusterLogForwarder** CR에서 다음 구조 유형을 사용하여 Elasticsearch 로그 저장소의 인덱스 이름을 구성할 수 있습니다.

- **structuredTypeKey** (문자열, 선택 사항)는 메시지 필드의 이름입니다. 해당 필드의 값은 인덱스 이름을 구성하는 데 사용됩니다.
  - **kubernetes.labels.<key>**는 인덱스 이름을 구성하는 데 사용되는 Kubernetes Pod 레이블입니다.
  - **openshift.labels.<key>**는 **ClusterLogForwarder** CR의 **pipeline.label.<key>** 요소이며 인덱스 이름을 구성하는 데 사용되는 값이 있습니다.
  - **kubernetes.container\_name**은 컨테이너 이름을 사용하여 인덱스 이름을 구성합니다.
- **structuredTypeName**: (문자열, 선택 사항) **structuredTypeKey**가 설정되지 않았거나 키가 없으면 OpenShift Logging은 구조화된 유형으로 **structuredTypeName** 값을 사용합니다. **structuredTypeKey** 및 **structuredTypeName**을 함께 사용하면 **structuredTypeKey**의 키가 JSON 로그 데이터에서 누락된 경우 **structuredTypeName**은 대체 인덱스 이름을 제공합니다.



### 참고

**structuredTypeKey** 값을 "Log Record Fields(로그 레코드 필드)" 항목에 표시된 모든 필드로 설정할 수 있지만 가장 유용한 필드가 앞의 구조 유형 목록에 표시됩니다.

### A structuredTypeKey: kubernetes.labels.<key> example

다음을 확인합니다.

- 클러스터에서 "apache" 및 "google"의 두 가지 형식으로 JSON 로그를 생성하는 애플리케이션 Pod를 실행하고 있습니다.
- 사용자는 **logFormat=apache** 및 **logFormat=google**를 사용하여 이러한 애플리케이션 pod에 레이블을 지정합니다.
- **ClusterLogForwarder** CR YAML 파일에서 다음 코드 조각을 사용합니다.

```
outputDefaults:
- elasticsearch:
  structuredTypeKey: kubernetes.labels.logFormat 1
  structuredTypeName: nologformat
pipelines:
```

```
- inputRefs: <application>
  outputRefs: default
  parse: json 2
```

- 1 Kubernetes **logFormat** 레이블로 구성된 키-값 쌍의 값을 사용합니다.
- 2 JSON 로그를 구문 분석할 수 있습니다.

이 경우 다음과 같은 구조화된 로그 레코드가 **app-apache-write** 인덱스로 이동합니다.

```
{
  "structured":{"name":"fred","home":"bedrock"},
  "kubernetes":{"labels":{"logFormat": "apache", ...}}
}
```

다음과 같은 구조화된 로그 레코드는 **app-google-write** 인덱스로 이동합니다.

```
{
  "structured":{"name":"wilma","home":"bedrock"},
  "kubernetes":{"labels":{"logFormat": "google", ...}}
}
```

A structuredTypeKey: openshift.labels.<key> example

ClusterLogForwarder CR YAML 파일에서 다음 코드 조각을 사용한다고 가정합니다.

```
outputDefaults:
- elasticsearch:
  structuredTypeKey: openshift.labels.myLabel 1
  structuredTypeName: nologformat
pipelines:
- name: application-logs
  inputRefs:
  - application
  - audit
  outputRefs:
  - elasticsearch-secure
  - default
  parse: json
  labels:
    myLabel: myValue 2
```

- 1 OpenShift **myLabel** 레이블로 구성된 키-값 쌍의 값을 사용합니다.
- 2 **myLabel** 요소는 구조화된 로그 레코드에 문자열 값 **myValue**를 제공합니다.

이 경우 다음과 같은 구조화된 로그 레코드가 **app-myValue-write** 인덱스로 이동합니다.

```
{
  "structured":{"name":"fred","home":"bedrock"},
  "openshift":{"labels":{"myLabel": "myValue", ...}}
}
```

## 추가 고려 사항

- 구조화된 레코드에 대한 Elasticsearch 인덱스는 구조화된 유형 앞에 "app-"를 추가하고 "-write"를 추가하여 구성됩니다.
- 구조화되지 않은 레코드는 구조화된 인덱스로 전송되지 않습니다. 애플리케이션, 인프라 또는 감사 인덱스에서 일반적으로 인덱싱됩니다.
- 비어 있지 않은 구조화된 유형이 없는 경우 **structured** 필드없이 *unstructured* 레코드를 전달합니다.

Elasticsearch에 너무 많은 인덱스로 과부하가 발생하지 않는 것이 중요합니다. 각 애플리케이션 또는 네임스페이스에는 별도의 구조화된 유형만 사용하는 것이 아니라 별도의 로그 형식에만 사용합니다. 예를 들어 대부분의 Apache 애플리케이션은 **LogApache**와 같은 동일한 JSON 로그 형식과 구조화된 유형을 사용합니다.

## 8.3. ELASTICSEARCH 로그 저장소로 JSON 로그 전달

Elasticsearch 로그 저장소의 경우 JSON 로그 항목이 *다른 스키마를 따르는 경우 ClusterLogForwarder* 사용자 정의 리소스(CR)를 구성하여 각 JSON 스키마를 단일 출력 정의로 그룹화합니다. 이렇게 하면 Elasticsearch는 각 스키마에 대해 별도의 인덱스를 사용합니다.



### 중요

동일한 인덱스로 다른 스키마를 전달하면 유형 충돌 및 카디널리티 문제가 발생할 수 있으므로 Elasticsearch 저장소로 데이터를 전달하기 전에 이 구성을 수행해야 합니다.

너무 많은 인덱스를 보유하는 것과 관련된 성능 문제를 방지하려면 공통 스키마로 표준화하여 가능한 스키마 수를 유지하는 것이 좋습니다.

## 절차

1. 다음 조각을 **ClusterLogForwarder** CR YAML 파일에 추가합니다.

```
outputDefaults:
- elasticsearch:
  structuredTypeKey: <log record field>
  structuredTypeName: <name>
pipelines:
- inputRefs:
- application
  outputRefs: default
  parse: json
```

2. 선택 사항: **structuredTypeKey** 를 사용하여 이전 항목에 설명된 대로 로그 레코드 필드 중 하나를 지정하고 **Elasticsearch에 대한 JSON 로그 데이터** 구성을 지정합니다. 그렇지 않으면 다음 행을 제거합니다.
3. 선택 사항: **structuredTypeName** 을 사용하여 이전 항목에 설명된 <name> 을 지정하고 **Elasticsearch에 대한 JSON 로그 데이터**를 구성합니다. 그렇지 않으면 다음 행을 제거합니다.



### 중요

JSON 로그를 구문 분석하려면 **structuredTypeKey** 또는 **structuredTypeName**, **structuredTypeKey** 및 **structuredTypeName** 모두를 설정해야 합니다.

4. **inputRefs**의 경우 **application, infrastructure**, 또는 **audit**등 해당 파이프라인을 사용하여 전달해야 하는 로그 유형을 지정합니다.
5. **parse: json** 요소를 파이프라인에 추가합니다.
6. CR 오브젝트를 생성합니다.

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

Red Hat OpenShift Logging Operator는 Fluentd Pod를 재배포합니다. Pod가 재배포되지 않으면 Fluentd Pod를 삭제하여 강제로 재배포할 수 있습니다.

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

#### 추가 리소스

- [타사 시스템에 로그 전달](#)

## 9장. 쿠버네티스 이벤트 수집 및 저장

OpenShift Container Platform 이벤트 라우터는 Kubernetes 이벤트를 감시하고 OpenShift Logging에 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.

### 9.1. 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.
- OpenShift Logging이 설치되어 있어야 합니다.

#### 프로세스

1. 이벤트 라우터용 템플릿을 생성합니다.

```
kind: Template
apiVersion: template.openshift.io/v1
metadata:
  name: eventrouter-template
  annotations:
    description: "A pod forwarding kubernetes events to OpenShift Logging stack."
    tags: "events,EFK,logging,cluster-logging"
objects:
- kind: ServiceAccount 1
  apiVersion: v1
  metadata:
    name: eventrouter
    namespace: ${NAMESPACE}
- kind: ClusterRole 2
  apiVersion: rbac.authorization.k8s.io/v1
  metadata:
    name: event-reader
  rules:
  - apiGroups: [""]
    resources: ["events"]
```

```
  verbs: ["get", "watch", "list"]
- kind: ClusterRoleBinding 3
  apiVersion: rbac.authorization.k8s.io/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 6
  displayName: Image
  value: "registry.redhat.io/openshift-logging/eventrouter-rhel8:v0.3"
- name: CPU 7
  displayName: CPU
  value: "100m"
- name: MEMORY 8
  displayName: Memory
  value: "128Mi"
- name: NAMESPACE
  displayName: Namespace
  value: "openshift-logging" 9

```

- 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 v0.3 과 같은 태그로 식별되는 이미지를 지정합니다.
- 7 이벤트 라우터 Pod에 할당할 최소 메모리 양을 지정합니다. 기본값은 128Mi입니다.
- 8 이벤트 라우터 Pod에 할당할 최소 CPU 양을 지정합니다. 기본값은 100m입니다.
- 9 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/eventrouter created
clusterrole.authorization.openshift.io/event-reader created
clusterrolebinding.authorization.openshift.io/event-reader-binding created

```

```
configmap/eventrouter created
deployment.apps/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.

## 10장. OPENSIFT LOGGING 업데이트

표 10.1. Red Hat OpenShift Logging (RHOL)에 대한 OpenShift Container Platform 버전 지원

	4.7	4.8	4.9
RHOL 5.1	X	X	
RHOL 5.2	X	X	X
RHOL 5.3		X	X

OpenShift Container Platform 4.6 및 이전 버전의 OpenShift Logging 5.x로 클러스터 로깅에서 업그레이드하려면 OpenShift Container Platform 클러스터를 4.7 또는 4.8로 업데이트합니다. 그런 다음 다음 operator를 업데이트합니다.

- Elasticsearch Operator 4.x에서 OpenShift Elasticsearch Operator 5.x로 업데이트
- Cluster Logging Operator 4.x에서 Red Hat OpenShift Logging Operator 5.x로 업데이트

이전 버전의 OpenShift Logging에서 현재 버전으로 업그레이드하려면 OpenShift Elasticsearch Operator 및 Red Hat OpenShift Logging Operator를 현재 버전으로 업데이트합니다.

### 10.1. OPENSIFT CONTAINER PLATFORM 4.6 또는 이전 버전의 OPENSIFT LOGGING 5.X로 클러스터 로깅에서 업데이트

OpenShift Container Platform 4.7에서는 다음과 같이 이름이 변경되었습니다.

- 클러스터 로깅 기능은 Red Hat OpenShift Logging 5.x 제품이 되었습니다.
- Cluster Logging Operator가 Red Hat OpenShift Logging Operator가 되었습니다.
- Elasticsearch Operator가 OpenShift Elasticsearch Operator가 되었습니다.

OpenShift Container Platform 4.6 및 이전 버전의 OpenShift Logging 5.x로 클러스터 로깅에서 업그레이드하려면 OpenShift Container Platform 클러스터를 4.7 또는 4.8로 업데이트합니다. 그런 다음 다음 operator를 업데이트합니다.

- Elasticsearch Operator 4.x에서 OpenShift Elasticsearch Operator 5.x로 업데이트
- Cluster Logging Operator 4.x에서 Red Hat OpenShift Logging Operator 5.x로 업데이트



#### 중요

Red Hat OpenShift Logging Operator를 업데이트하기 전에 OpenShift Elasticsearch Operator를 업데이트해야 합니다. 두 Operator를 동일한 버전으로 업데이트해야 합니다.

Operator를 잘못된 순서로 업데이트하면 Kibana가 업데이트되지 않고 Kibana 사용자 지정 리소스(CR)가 생성되지 않습니다. 이 문제를 해결하려면 Red Hat OpenShift Logging Operator Pod를 삭제합니다. Red Hat OpenShift Logging Operator Pod가 재배포되면 Kibana CR을 생성하고 Kibana를 다시 사용할 수 있게 됩니다.

사전 요구 사항

- OpenShift Container Platform 4.7 이상 버전이어야 합니다.
- OpenShift Logging 상태가 정상이어야 합니다.
  - 모든 Pod가 **ready** 상태입니다.
  - Elasticsearch 클러스터는 정상입니다.
- Elasticsearch 및 Kibana 데이터가 백업됩니다.

절차

1. OpenShift Elasticsearch Operator를 업데이트합니다.
  - a. 웹 콘솔에서 Operator → 설치된 Operator를 클릭합니다.
  - b. **openshift-operators-redhat** 프로젝트를 선택합니다.
  - c. OpenShift Elasticsearch Operator를 클릭합니다.
  - d. 서브스크립션 → 채널을 클릭합니다.
  - e. 서브스크립션 업데이트 채널 변경 창에서 5.0 또는 stable-5.1을 선택하고 저장을 클릭합니다.
  - f. 몇 초 정도 기다린 후 Operator → 설치된 Operator를 클릭합니다.  
OpenShift Elasticsearch Operator 버전이 5.x.x인지 확인합니다.  
  
상태 필드가 성공으로 표시될 때까지 기다립니다.
2. Cluster Logging Operator 업데이트:
  - a. 웹 콘솔에서 Operator → 설치된 Operator를 클릭합니다.
  - b. **openshift-logging** 프로젝트를 선택합니다.
  - c. Cluster Logging Operator를 클릭합니다.
  - d. 서브스크립션 → 채널을 클릭합니다.
  - e. 서브스크립션 업데이트 채널 변경 창에서 5.0 또는 stable-5.1을 선택하고 저장을 클릭합니다.
  - f. 몇 초 정도 기다린 후 Operator → 설치된 Operator를 클릭합니다.  
Red Hat OpenShif Logging Operator 버전이 5.0.x 또는 5.1.x인지 확인합니다.  
  
상태 필드가 성공으로 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 -- 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
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 5.0 또는 5.1로 업데이트되고 인덱스가 **green**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 10.1. 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
1SJdCqIzTPWIIAaOUd78yg 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_GVQVQZ6Yh9lg 3 1    223788    0    292    146
green open  infra-000015
JRBbAbEmSMqK5X40df9HbQ 3 1    224371    0    291    145
green open  .orphaned.2020.06.30
```

n_xQC2dWQzConkvQqei3YA	3	1	9	0	0	0	
green open infra-000007							
llkkAVSzSOMosWTSAJM_hg	3	1	228584	0	296	148	
green open infra-000005							
d9BoGQdiQASsS3BBFm2iRA	3	1	227987	0	297	148	
green open infra-000003						1-	
goREK1QUKIQAIVkWWaQ	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-	
bSxSWR5xYZB6IVg	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						KR9mMFUpQI-	
jaYtanyIGw	3	1	228166	0	298	148	
green open audit-000001							
eERqLdLmQOiQDFES1LBATQ	3	1	0	0	0	0	

e. 로그 수집기가 5.0 또는 5.1로 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 5.0 또는 5.1로 업데이트되었는지 확인합니다.

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

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

예 10.2. 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
}
]

```

## 10.2. OPENSIFT LOGGING을 현재 버전으로 업데이트

OpenShift Logging을 5.x에서 현재 버전으로 업데이트하려면 OpenShift Elasticsearch Operator 및 Red Hat OpenShift Logging Operator의 서브스크립션을 변경합니다.



### 중요

Red Hat OpenShift Logging Operator를 업데이트하기 전에 OpenShift Elasticsearch Operator를 업데이트해야 합니다. 두 Operator를 동일한 버전으로 업데이트해야 합니다.

Operator를 잘못된 순서로 업데이트하면 Kibana가 업데이트되지 않고 Kibana 사용자 지정 리소스(CR)가 생성되지 않습니다. 이 문제를 해결하려면 Red Hat OpenShift Logging Operator Pod를 삭제합니다. Red Hat OpenShift Logging Operator Pod가 재배포되면 Kibana CR을 생성하고 Kibana를 다시 사용할 수 있게 됩니다.

### 사전 요구 사항

- OpenShift Container Platform 4.7 이상 버전이어야 합니다.
- OpenShift Logging 상태가 정상이어야 합니다.
  - 모든 Pod가 **ready** 상태입니다.
  - Elasticsearch 클러스터는 정상입니다.
- Elasticsearch 및 Kibana 데이터가 백업됩니다.

### 절차

1. OpenShift Elasticsearch Operator를 업데이트합니다.
  - a. 웹 콘솔에서 Operator → 설치된 Operator를 클릭합니다.
  - b. **openshift-operators-redhat** 프로젝트를 선택합니다.
  - c. OpenShift Elasticsearch Operator를 클릭합니다.
  - d. 서브스크립션 → 채널을 클릭합니다.
  - e. 서브스크립션 업데이트 채널 변경 창에서 **stable-5.x**을 선택하고 저장을 클릭합니다.
  - f. 몇 초 정도 기다린 후 Operator → 설치된 Operator를 클릭합니다.  
OpenShift Elasticsearch Operator 버전이 5.x.x인지 확인합니다.  
  
상태 필드가 성공으로 표시될 때까지 기다립니다.

2. Red Hat OpenShift Logging Operator를 업데이트합니다.
  - a. 웹 콘솔에서 Operator → 설치된 Operator를 클릭합니다.
  - b. **openshift-logging** 프로젝트를 선택합니다.
  - c. Red Hat OpenShift Logging Operator를 클릭합니다.
  - d. 서브스크립션 → 채널을 클릭합니다.
  - e. 서브스크립션 업데이트 채널 변경 창에서 **stable-5.x**을 선택하고 저장을 클릭합니다.
  - f. 몇 초 정도 기다린 후 Operator → 설치된 Operator를 클릭합니다.  
Red Hat OpenShif Logging Operator 버전이 5.x.x인지 확인합니다.  
  
상태 필드가 성공으로 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 -- 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
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 5.x로 업데이트되고 인덱스가 green인지 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 10.3. 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
1SJdCqIZTPWIIAaOUd78yg 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_GVQVQZ6Yh9lg 3 1    223788    0    292    146
green open  infra-000015
JRBbAbEmSMqK5X40df9HbQ 3 1    224371    0    291    145
green open  .orphaned.2020.06.30
n_xQC2dWQzConkvQqei3YA 3 1     9         0    0       0
green open  infra-000007
llkkAVSszSOMosWTSAJM_hg 3 1    228584    0    296    148
green open  infra-000005
d9BoGQdiQASsS3BBFm2iRA 3 1    227987    0    297    148
green open  infra-000003
goREK1QUKIQPAIVkWVaQ 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
KBSXGQKiO7hdapDE23g 3 1    226676    0    295    147
green open  infra-000001
```

bSxSWR5xYZB6IVg	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 5.x로 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 5.x로 업데이트되었는지 확인합니다.

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

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

예 10.4. 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  
}  
]
```

## 11장. 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.

### 11.1. ELASTISEARCH 및 OPENSIFT 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 → 대시보드를 클릭합니다.
2. 대시보드 페이지의 대시보드 메뉴에서 로깅/Elasticsearch 노드 또는 OpenShift 로깅 을 선택합니다.  
로깅/Elasticsearch 노드 대시보드의 경우 보려는 Elasticsearch 노드를 선택하고 데이터 해상도를 설정할 수 있습니다.

여러 데이터 차트를 보여주는 적절한 대시보드가 표시됩니다.

3. 선택적으로 시간 범위 및 새로 고침 간격 메뉴에서 데이터를 표시하거나 새로 고칠 다른 시간 범위를 선택합니다.



### 참고

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

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

### 11.2. OPENSIFT 로깅 대시보드 정보

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

표 11.1. OpenShift 로깅 차트

지표	설명
Elastic 클러스터 상태	<p>현재 Elasticsearch 상태:</p> <ul style="list-style-type: none"> <li>● 온라인 - Elasticsearch 인스턴스가 온라인 상태를 나타냅니다.</li> <li>● 오프라인 - Elasticsearch 인스턴스가 오프라인 상태를 나타냅니다.</li> </ul>
Elastic 노드	Elasticsearch 인스턴스의 총 Elasticsearch 노드 수입니다.
Elastic Shard	Elasticsearch 인스턴스의 총 Elasticsearch shard 수입니다.
Elastic 문서	Elasticsearch 인스턴스의 총 Elasticsearch 문서 수입니다.
디스크의 총 인덱스 크기	Elasticsearch 인덱스에 사용 중인 총 디스크 공간입니다.
Elastic 보류 작업	인덱스 생성, 인덱스 매핑, shard 할당 또는 shard 오류와 같이 완료되지 않은 Elasticsearch 변경의 총 수입니다.
Elastic JVM GC 시간	JVM이 클러스터에서 Elasticsearch 가비지 수집 작업을 실행하는 데 소비한 시간입니다.
Elastic JVM GC 속도	JVM이 초당 가비지 활동을 실행한 총 횟수입니다.
Elastic 쿼리/가져오기 대기 시간 합계	<ul style="list-style-type: none"> <li>● 쿼리 대기 시간: 각 Elasticsearch 검색 쿼리를 실행하는 데 걸리는 평균 시간입니다.</li> <li>● 가져오기 대기 시간: 각 Elasticsearch 검색 쿼리에서 데이터를 가져오는 데 걸리는 평균 시간입니다.</li> </ul> <p>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.</p>
Elastic 쿼리 속도	각 Elasticsearch 노드에 대해 Elasticsearch 인스턴스에 대해 실행된 초당 총 쿼리입니다.
CPU	Elasticsearch, Fluentd 및 Kibana에서 사용하는 CPU 양(각 구성 요소에 대해 표시됨).

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

### 11.3. 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 11.2. Elasticsearch 상태 필드

지표	설명
----	----

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

### Elasticsearch 클러스터 인덱스 shard 상태

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

인덱스 shard는 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 11.3. Elasticsearch 클러스터 shard 상태 차트

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

**Elasticsearch 노드 지표**

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

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

표 11.4. Elasticsearch 노드 지표 차트

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

지표	설명
인덱싱 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 11.5. 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 11.6. Elasticsearch 노드 쿼리 차트

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

**Elasticsearch 인덱스 제한**

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

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

표 11.7. 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 11.8. JVM 힙 통계 차트

지표	설명
사용된 힙	선택한 Elasticsearch 노드에서 사용되는 총 할당된 JVM 힙 공간의 양입니다.
GC 수	오래된 가비지 수집에 의해 선택된 Elasticsearch 노드에서 실행된 가비지 수집 작업의 수입입니다.
GC 시간	JVM이 선택한 Elasticsearch 노드에서 가비지 수집 작업을 실행하는 데 소비한 시간(오래된 가비지 및 새 가비지 수집 기준)입니다.

## 12장. 로깅 문제 해결

### 12.1. OPENSIFT LOGGING 상태 보기

Red Hat OpenShift Logging Operator 및 여러 OpenShift Logging 구성 요소의 상태를 볼 수 있습니다.

#### 12.1.1. Red Hat OpenShift Logging Operator의 상태 보기

Red Hat OpenShift Logging Operator의 상태를 볼 수 있습니다.

사전 요구 사항

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

1. `openshift-logging` 프로젝트로 변경합니다.

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

2. OpenShift Logging 상태를 보려면 다음을 수행합니다.

- a. OpenShift Logging 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 스텐자에 나타납니다.
- ② Fluentd Pod에 대한 정보.
- ③ Elasticsearch 클러스터 건강, 녹색, 노란색 또는 빨간색을 포함한 Elasticsearch Pod에 대한 정보입니다.
- ④ Kibana Pod에 대한 정보.

### 12.1.1.1. 상태 메시지 예

다음은 OpenShift Logging 인스턴스의 **Status.Nodes** 섹션에 있는 일부 상태 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:

## 12.1.2. OpenShift Logging 구성 요소의 상태 보기

여러 OpenShift Logging 구성 요소의 상태를 볼 수 있습니다.

사전 요구 사항

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

1. openshift-logging 프로젝트로 변경합니다.

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

2. OpenShift Logging 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. OpenShift Logging 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----

```

## 12.2. 로그 저장소의 상태 보기

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

### 12.2.1. 로그 저장소의 상태 보기

로그 저장소의 상태를 볼 수 있습니다.

사전 요구 사항

- OpenShift Logging 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, 노란색입니다.
  - 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 `ready` 상태 아래에 나열된 클러스터의 로그를 저장 클라이언트, 데이터 및 마스터 Pod.

### 12.2.1.1. 상태 메시지 예

다음은 Elasticsearch 인스턴스의 상태 섹션에 있는 일부 조건 메시지의 예입니다.

다음 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: {}

```

다음 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: {}

```

다음 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: Unscheduleable
        status: "True"
        type: Unscheduleable

```

다음 상태 메시지는 로그 저장소 CR에서 PVC(영구 볼륨 클레임)가 존재하지 않음을 나타냅니다.

```

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

```

다음 상태 메시지는 로그 저장소 클러스터에 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
```

다음 상태 메시지는 Elasticsearch 스토리지가 변경 작업을 지원하지 않음을 나타냅니다.

예를 들면 다음과 같습니다.

```
status:
  clusterHealth: green
  conditions:
    - lastTransitionTime: "2021-05-07T01:05:13Z"
      message: Changing the storage structure for a custom resource is not supported
      reason: StorageStructureChangelgnored
      status: 'True'
      type: StorageStructureChangelgnored
```

reason 및 type 필드는 지원되지 않는 변경 유형을 지정합니다.

### StorageClassNameChangelgnored

스토리지 클래스 이름에 대한 지원되지 않는 변경 사항입니다.

### StorageSizeChangelgnored

스토리지 크기에 대한 지원되지 않는 변경 사항입니다.

### StorageStructureChangelgnored

임시 스토리지 구조와 영구저장장치 구조 간에는 지원되지 않는 변경 사항입니다.



#### 중요

임시 스토리지에서 영구 스토리지로 전환하도록 **ClusterLogging** 사용자 정의 리소스 (CR)를 구성하려는 경우 OpenShift Elasticsearch Operator는 PVC(영구 볼륨 클레임)를 생성하지만 PV(영구 볼륨)를 생성하지 않습니다. **StorageStructureChangelgnored** 상태를 지우려면 **ClusterLogging** CR로 변경 사항을 취소하고 PVC를 삭제해야 합니다.

## 12.2.2. 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
1 1 5 0 0 0 iDjSCH7aSUGhldq0LheLBQ
green open .kibana_-377444158_kubeadmin
yBywZ9GfSrKebz5gWBZbjw 3 1 1 0 0 0
green open infra-000001
z6Dpe__ORgiopEpW6YI44A 3 1 871000 0 874 436
green open app-000001
3 1 2453 0 3 1 hlrazQCeSISewG3c2VlvsQ
green open .kibana_1
1 1 0 0 0 0 JCitcBMSQxKOvlq6iQW6wg
green open .kibana_-1595131456_user1
ka0W3okS-mQ 3 1 1 0 0 0 gIYFIEGRRe-
```

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/openshift-logging/elasticsearch6-rhel8
    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/openshift-logging/elasticsearch6-
          rhel8@sha256:4265742c7cdd85359140e2d7d703e4311b6497eec7676957f455d6908e7
          b1c25
    Readiness: exec [/usr/share/elasticsearch/probe/readiness.sh] delay=10s
              timeout=30s period=5s #success=1 #failure=3
```

```
....
Events:      <none>
```

### 12.3. OPENSIFT LOGGING 경고 이해

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

#### 12.3.1. 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

OpenShift Logging 및 기타 OpenShift Container Platform 경고를 보려면 다음을 수행합니다.

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

#### 추가 리소스

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

#### 12.3.2. 로깅 수집기 경고 정보

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

표 12.1. Fluentd Prometheus 경고

경고	메시지	설명	심각도
<b>FluentDHighErrorRate</b>	<b>fluentd &lt;instance&gt;에 의해 레코드의 &lt;value&gt;에서 오류가 발생했습니다.</b>	FluentD 출력 오류의 수는 높으며 기본적으로 이전 15분 동안 10개 이상입니다.	경고
<b>FluentdNodeDown</b>	<b>Prometheus는 fluentd &lt;instance&gt;를 10분 이상 스크랩할 수 없습니다.</b>	Fluentd는 Prometheus가 특정 Fluentd 인스턴스를 스크랩할 수 없다고 보고했습니다.	심각

경고	메시지	설명	심각도
<b>FluentdQueueLengthIncreasing</b>	지난 12시간 동안 <b>fluentd &lt;instance&gt;</b> 버퍼 큐 길이는 1보다 지속적으로 증가했습니다. 현재 값은 <b>&lt;value&gt;</b> 입니다.	Fluentd는 큐 크기가 증가하고 있다고 보고합니다.	심각
<b>FluentDVeryHighErrorRate</b>	<b>fluentd &lt;instance&gt;</b> 에 의해 레코드의 <b>&lt;value&gt;</b> 에서 오류가 발생했습니다.	FluentD 출력 오류의 수는 기본적으로 이전 15분 동안 25개 이상으로 매우 높습니다.	심각

### 12.3.3. Elasticsearch 경고 규칙 정보

이러한 경고 규칙을 Prometheus에서 볼 수 있습니다.

표 12.2. 경고 규칙

경고	설명	심각도
<b>ElasticsearchClusterNotHealthy</b>	클러스터 상태가 2분 이상 빨간색이었습니다. 클러스터가 쓰기를 허용하지 않거나 shard가 누락되었거나 마스터 노드가 아직 선택되지 않았을 수 있습니다.	심각
<b>ElasticsearchClusterNotHealthy</b>	클러스터 상태가 최소 20분 동안 노란색이었습니다. 일부 shard 복제본이 할당되지 않았습니다.	경고
<b>ElasticsearchDiskSpaceRunningLow</b>	클러스터는 향후 6시간 내에 디스크 공간이 부족할 것으로 예상됩니다.	심각
<b>ElasticsearchHighFileDescriptorUsage</b>	클러스터는 다음 시간 내에 파일 설명자가 없을 것으로 예상됩니다.	경고
<b>ElasticsearchJVMHeapUseHigh</b>	지정된 노드의 JVM 힙 사용량이 높습니다.	경고
<b>ElasticsearchNodeDiskWatermarkReached</b>	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
<b>ElasticsearchNodeDiskWatermarkReached</b>	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
<b>ElasticsearchNodeDiskWatermarkReached</b>	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

경고	설명	심각도
<b>ElasticsearchJVMHeapUseHigh</b>	지정된 노드의 JVM 힙 사용량이 너무 높습니다.	경고
<b>ElasticsearchWriteRequestsRejectionJumps</b>	Elasticsearch의 지정된 노드에서 쓰기 거부가 증가하고 있습니다. 이 노드는 인덱싱 속도를 따라가지 못할 수 있습니다.	경고
<b>AggregatedLoggingSystemCPULHigh</b>	지정된 노드의 시스템에서 사용하는 CPU가 너무 높습니다.	경고
<b>ElasticsearchProcessCPUHigh</b>	지정된 노드에서 Elasticsearch가 사용하는 CPU가 너무 높습니다.	경고

## 12.4. 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 OpenShift Logging 구성 요소에 대한 진단 정보를 수집할 수 있습니다.

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



### 참고

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

### 12.4.1. must-gather 툴 정보

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

OpenShift Logging 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, 로그 시각화 프로그램의 상태를 포함하여 **openshift-logging** 및 **openshift-operators-redhat** 네임스페이스의 OpenShift Logging 리소스

**oc adm must-gather**를 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.

### 12.4.2. 사전 요구 사항

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

### 12.4.3. OpenShift Logging 데이터 수집

`oc adm must-gather` CLI 명령을 사용하여 OpenShift Logging 환경에 대한 정보를 수집할 수 있습니다.

프로세스

`must-gather`로 OpenShift Logging 정보를 수집하려면 다음을 수행하십시오.

1. `must-gather` 정보를 저장하려는 디렉터리로 이동합니다.
2. OpenShift Logging 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` 틀에서 현재 디렉터리 내에 `must-gather.local`로 시작하는 새 디렉터를 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.

## 12.5. 심각한 경고 문제 해결

### 12.5.1. Elasticsearch 클러스터 상태가 빨간색임

하나 이상의 기본 shard와 해당 복제본이 노드에 할당되지 않습니다.

문제 해결

1. Elasticsearch 클러스터 상태를 확인하고 클러스터 `status`가 빨간색인지 확인합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- health
```

2. 클러스터에 참여한 노드를 나열합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --query=_cat/nodes?v
```

3. Elasticsearch pod를 나열하고 이전 단계의 명령 출력의 노드와 비교합니다.

```
oc -n openshift-logging get pods -l component=elasticsearch
```

4. 일부 Elasticsearch 노드가 클러스터에 참여하지 않은 경우 다음 단계를 수행합니다.

- a. Elasticsearch에 선택한 컨트롤 플레인 노드가 있는지 확인합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --query=_cat/master?v
```

- b. 선택한 컨트롤 플레인 노드의 Pod 로그를 검토하여 문제가 있는지 확인합니다.

```
oc logs <elasticsearch_master_pod_name> -c elasticsearch -n openshift-logging
```

- c. 클러스터에 참여하지 않은 노드의 로그에서 문제가 있는지 검토합니다.

```
oc logs <elasticsearch_node_name> -c elasticsearch -n openshift-logging
```

5. 모든 노드가 클러스터에 참여한 경우 다음 단계를 수행하여 클러스터가 복구 프로세스 중인지 확인합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --query=_cat/recovery?active_only=true
```

명령 출력이 없는 경우 복구 프로세스가 보류 중인 작업에서 지연되거나 중단될 수 있습니다.

6. 보류 중인 작업이 있는지 확인합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- health |grep number_of_pending_tasks
```

7. 보류 중인 작업이 있는 경우 상태를 모니터링합니다.

상태가 변경되고 클러스터가 복구 중임을 나타내는 경우 계속 대기합니다. 복구 시간은 클러스터의 크기와 기타 요인에 따라 다릅니다.

그렇지 않으면 보류 중인 작업의 상태가 변경되지 않는 경우 복구가 중지되었음을 나타냅니다.

8. 복구가 중단된 것처럼 보이는 경우 `cluster.routing.allocation.enable`이 `none`으로 설정되어 있는지 확인합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --query=_cluster/settings?pretty
```

9. `cluster.routing.allocation.enable`이 `none`으로 설정되어 있으면 `all`로 설정합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --query=_cluster/settings?pretty -X PUT -d '{"persistent": {"cluster.routing.allocation.enable": "all"}}'
```

10. 어떤 인덱스가 아직 빨간색인지 확인합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --query=_cat/indices?v
```

11. 인덱스가 빨간색이면 다음 단계를 수행하여 지웁니다.

- a. 캐시를 지웁니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --query=<elasticsearch_index_name>/_cache/clear?pretty
```

- b. 최대 할당 채시도 횟수를 늘립니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util
--query=<elasticsearch_index_name>/_settings?pretty -X PUT -d
'{"index.allocation.max_retries":10}'
```

- c. 모든 스크롤 항목을 삭제합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util
--query=_search/scroll/_all -X DELETE
```

- d. 시간 제한을 늘립니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util
--query=<elasticsearch_index_name>/_settings?pretty -X PUT -d
'{"index.unassigned.node_left.delayed_timeout":"10m"}'
```

12. 이전 단계에서 빨간색 인덱스를 지우지 않으면 인덱스를 개별적으로 삭제합니다.

- a. 빨간색 인덱스 이름을 확인합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util
--query=_cat/indices?v
```

- b. 빨간색 인덱스를 삭제합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util
--query=<elasticsearch_red_index_name> -X DELETE
```

13. 빨간색 인덱스가 없고 클러스터 상태가 빨간색이면 데이터 노드에서 지속적으로 처리 로드가 높은 지 확인합니다.

- a. Elasticsearch JVM 힙 사용량이 높은지 확인합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util
--query=_nodes/stats?pretty
```

명령 출력에서 `node_name.jvm.mem.heap_used_percent` 필드를 검토하여 JVM 힙 사용량을 확인합니다.

- b. CPU 사용률이 높은지 확인합니다.

#### 추가 리소스

- Elasticsearch 주제에서 "Free up or increase disk space"를 검색하고 **빨간색 또는 노란색 클러스터 상태를 수정합니다.**

### 12.5.2. Elasticsearch 클러스터 상태가 노란색임

하나 이상의 기본 shard의 복제본 shard는 노드에 할당되지 않습니다.

#### 문제 해결

1. ClusterLogging CR에서 `nodeCount`를 조정하여 노드 수를 늘립니다.

→ → → →

## 추가 리소스

- [클러스터 로깅 사용자 정의 리소스 정보](#)
- [로그 저장소에 대한 영구 스토리지 구성](#)
- Elasticsearch 주제에서 "Free up or increase disk space"를 검색하고 [빨간색 또는 노란색 클러스터 상태를 수정합니다.](#)

## 12.5.3. Elasticsearch 노드 디스크 Low Watermark Reached

Elasticsearch는 [낮은 워터마크에 도달하는 노드에 shard를 할당하지 않습니다.](#)

## 문제 해결

1. Elasticsearch가 배포된 노드를 식별합니다.

```
oc -n openshift-logging get po -o wide
```

2. unassigned shards가 있는지 확인합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
query=_cluster/health?pretty | grep unassigned_shards
```

3. 할당되지 않은 shard가 있는 경우 각 노드에서 디스크 공간을 확인합니다.

```
for pod in `oc -n openshift-logging get po -l component=elasticsearch -o
jsonpath='{.items[*].metadata.name}'`; do echo $pod; oc -n openshift-logging exec -c
elasticsearch $pod -- df -h /elasticsearch/persistent; done
```

4. nodes.node\_name.fs 필드를 확인하여 해당 노드에서 사용 가능한 디스크 공간을 확인합니다. 사용된 디스크 백분율이 85%를 초과하는 경우 노드가 낮은 워터마크를 초과하여 더 이상 이 노드에 shard를 할당할 수 없습니다.
5. 모든 노드의 디스크 공간을 늘리십시오.
6. 디스크 공간을 늘릴 수 없는 경우 클러스터에 새 데이터 노드를 추가해 보십시오.
7. 새 데이터 노드를 추가하는 데 문제가 있는 경우 전체 클러스터 중복 정책을 줄입니다.
  - a. 현재 redundancyPolicy를 확인합니다.

```
oc -n openshift-logging get es elasticsearch -o
jsonpath='{.spec.redundancyPolicy}'
```



## 참고

ClusterLogging CR을 사용하는 경우 다음을 입력합니다.

```
oc -n openshift-logging get cl -o
jsonpath='{.items[*].spec.logStore.elasticsearch.redundancyPolicy}'
```

- b. 클러스터 **redundancyPolicy**가 **SingleRedundancy** 보다 큰 경우 **SingleRedundancy**로 설정하고 이러한 변경 사항을 저장합니다.
8. 이전 단계에서 문제가 해결되지 않으면 이전 인덱스를 삭제합니다.
  - a. Elasticsearch의 모든 인덱스의 상태를 확인합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- indices
```

- b. 삭제할 수 있는 이전 인덱스를 확인합니다.
- c. 인덱스를 삭제합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --query=<elasticsearch_index_name> -X DELETE
```

#### 추가 리소스

- [클러스터 로깅 사용자 정의 리소스 정보](#)의 "샘플 ClusterLogging 사용자 정의 리소스 (CR)"에서 "redundancyPolicy"를 검색합니다.

### 12.5.4. Elasticsearch 노드 디스크 High Watermark Reached

Elasticsearch는 **높은 워터마크에 도달**한 노드에서 shard를 재배치하려고 합니다.

#### 문제 해결

1. Elasticsearch가 배포된 노드를 식별합니다.

```
oc -n openshift-logging get po -o wide
```

2. 각 노드의 디스크 공간을 확인합니다.

```
for pod in `oc -n openshift-logging get po -l component=elasticsearch -o jsonpath='{.items[*].metadata.name}'`; do echo $pod; oc -n openshift-logging exec -c elasticsearch $pod -- df -h /elasticsearch/persistent; done
```

3. 클러스터가 재조정 중인지 확인합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --query=_cluster/health?pretty | grep relocating_shards
```

명령 출력에 shard 재배치가 표시되면 High QCOWmark가 초과된 것입니다. High QCOWmark의 기본값은 90%입니다.

shard는 워터마크 임계값 제한을 넘지 않은 디스크 사용량이 낮은 노드로 재배치됩니다.

4. 특정 노드에 shard를 할당하려면 일부 공간을 확보합니다.
5. 모든 노드의 디스크 공간을 늘리십시오.
6. 디스크 공간을 늘릴 수 없는 경우 클러스터에 새 데이터 노드를 추가해 보십시오.

7. 새 데이터 노드를 추가하는 데 문제가 있는 경우 전체 클러스터 중복 정책을 줄입니다.

- a. 현재 **redundancyPolicy**를 확인합니다.

```
oc -n openshift-logging get es elasticsearch -o
jsonpath='{.spec.redundancyPolicy}'
```



참고

**ClusterLogging** CR을 사용하는 경우 다음을 입력합니다.

```
oc -n openshift-logging get cl -o
jsonpath='{.items[*].spec.logStore.elasticsearch.redundancyPolicy}'
```

- b. 클러스터 **redundancyPolicy**가 **SingleRedundancy** 보다 큰 경우 **SingleRedundancy**로 설정하고 이러한 변경 사항을 저장합니다.

8. 이전 단계에서 문제가 해결되지 않으면 이전 인덱스를 삭제합니다.

- a. Elasticsearch의 모든 인덱스의 상태를 확인합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> --
indices
```

- b. 삭제할 수 있는 이전 인덱스를 확인합니다.

- c. 인덱스를 삭제합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util
--query=<elasticsearch_index_name> -X DELETE
```

추가 리소스

- 클러스터 로깅 사용자 정의 리소스 정보의 "샘플 ClusterLogging 사용자 정의 리소스 (CR)"에서 "redundancyPolicy"를 검색합니다.

### 12.5.5. Elasticsearch 노드 디스크 Flood Watermark Reached

Elasticsearch는 이러한 두 조건을 모두 충족하는 모든 인덱스에 읽기 전용 인덱스 블록을 적용합니다.

- 하나 이상의 shard가 노드에 할당됩니다.
- 하나 이상의 디스크가 **플러드 단계**를 초과합니다.

문제 해결

1. Elasticsearch 노드의 디스크 공간을 확인합니다.

```
for pod in `oc -n openshift-logging get po -l component=elasticsearch -o
jsonpath='{.items[*].metadata.name}'`; do echo $pod; oc -n openshift-logging exec -c
elasticsearch $pod -- df -h /elasticsearch/persistent; done
```

**nodes.node\_name.fs** 필드를 확인하여 해당 노드에서 사용 가능한 디스크 공간을 확인합니다.

2. 사용된 디스크 백분율이 95%를 초과하면 노드가 플러드 워터마크를 초과했음을 나타냅니다. 이 특정 노드에 할당된 shard에 대해 쓰기가 차단됩니다.
3. 모든 노드의 디스크 공간을 늘리십시오.
4. 디스크 공간을 늘릴 수 없는 경우 클러스터에 새 데이터 노드를 추가해 보십시오.
5. 새 데이터 노드를 추가하는 데 문제가 있는 경우 전체 클러스터 중복 정책을 줄입니다.
  - a. 현재 **redundancyPolicy**를 확인합니다.

```
oc -n openshift-logging get es elasticsearch -o
jsonpath='{.spec.redundancyPolicy}'
```



참고

**ClusterLogging** CR을 사용하는 경우 다음을 입력합니다.

```
oc -n openshift-logging get cl -o
jsonpath='{.items[*].spec.logStore.elasticsearch.redundancyPolicy}'
```

- b. 클러스터 **redundancyPolicy**가 **SingleRedundancy** 보다 큰 경우 **SingleRedundancy**로 설정하고 이러한 변경 사항을 저장합니다.
6. 이전 단계에서 문제가 해결되지 않으면 이전 인덱스를 삭제합니다.
  - a. Elasticsearch의 모든 인덱스의 상태를 확인합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> --
indices
```

- b. 삭제할 수 있는 이전 인덱스를 확인합니다.
  - c. 인덱스를 삭제합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util
--query=<elasticsearch_index_name> -X DELETE
```

7. 사용된 디스크 공간이 90% 미만으로 줄어들 때까지 디스크 공간을 계속 확보하고 모니터링합니다. 그런 다음 이 특정 노드에 대한 쓰기 차단을 해제합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
query=_all/_settings?pretty -X PUT -d '{"index.blocks.read_only_allow_delete": null}'
```

#### 추가 리소스

- [클러스터 로깅 사용자 정의 리소스 정보](#)의 "샘플 **ClusterLogging** 사용자 정의 리소스 (CR)"에서 "redundancyPolicy"를 검색합니다.

#### 12.5.6. Elasticsearch JVM 힙 사용량이 높음

사용된 Elasticsearch 노드 JVM 힙 메모리는 75% 이상입니다.

## 문제 해결

힙 크기를 늘리는 것이 좋습니다.

### 12.5.7. 집계된 로깅 시스템 CPU가 높음

노드의 시스템 CPU 사용량이 높습니다.

## 문제 해결

클러스터 노드의 CPU를 확인합니다. 더 많은 CPU 리소스를 노드에 할당하는 것이 좋습니다.

### 12.5.8. Elasticsearch 프로세스 CPU가 높음

노드의 Elasticsearch 프로세스 CPU 사용량이 높습니다.

## 문제 해결

클러스터 노드의 CPU를 확인합니다. 더 많은 CPU 리소스를 노드에 할당하는 것이 좋습니다.

### 12.5.9. Elasticsearch 디스크 공간이 부족

Elasticsearch 클러스터는 현재 디스크 사용량에 따라 향후 6시간 이내에 디스크 공간이 부족해질 것으로 예상됩니다.

## 문제 해결

1. Elasticsearch 노드의 디스크 공간을 가져옵니다.

```
for pod in `oc -n openshift-logging get po -l component=elasticsearch -o jsonpath='{.items[*].metadata.name}'`; do echo $pod; oc -n openshift-logging exec -c elasticsearch $pod -- df -h /elasticsearch/persistent; done
```

2. 명령 출력에서 `nodes.node_name.fs` 필드를 확인하여 해당 노드의 사용 가능한 디스크 공간을 확인합니다.
3. 모든 노드의 디스크 공간을 늘리십시오.
4. 디스크 공간을 늘릴 수 없는 경우 클러스터에 새 데이터 노드를 추가해 보십시오.
5. 새 데이터 노드를 추가하는 데 문제가 있는 경우 전체 클러스터 중복 정책을 줄입니다.
  - a. 현재 `redundancyPolicy`를 확인합니다.

```
oc -n openshift-logging get es elasticsearch -o jsonpath='{.spec.redundancyPolicy}'
```



## 참고

ClusterLogging CR을 사용하는 경우 다음을 입력합니다.

```
oc -n openshift-logging get cl -o jsonpath='{.items[*].spec.logStore.elasticsearch.redundancyPolicy}'
```

- b. 클러스터 **redundancyPolicy**가 **SingleRedundancy** 보다 큰 경우 **SingleRedundancy**로 설정하고 이러한 변경 사항을 저장합니다.
6. 이전 단계에서 문제가 해결되지 않으면 이전 인덱스를 삭제합니다.
    - a. Elasticsearch의 모든 인덱스의 상태를 확인합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- indices
```

- b. 삭제할 수 있는 이전 인덱스를 확인합니다.
- c. 인덱스를 삭제합니다.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --query=<elasticsearch_index_name> -X DELETE
```

#### 추가 리소스

- [클러스터 로깅 사용자 정의 리소스 정보](#)의 "샘플 ClusterLogging 사용자 정의 리소스 (CR)"에서 "redundancyPolicy"를 검색합니다.
- [Elasticsearch 경고 규칙](#)에서 "ElasticsearchDiskSpaceRunningLow"를 검색합니다.
- Elasticsearch 주제에서 "Free up or increase disk space"를 검색하고 **빨간색 또는 노란색 클러스터 상태를 수정**합니다.

### 12.5.10. Elasticsearch FileDescriptor 사용량이 높음

현재 사용 추세를 기준으로 노드의 예상 파일 설명자 수가 충분하지 않습니다.

#### 문제 해결

Elasticsearch [파일 설명자](#) 항목에 설명된 대로 필요에 따라 각 노드의 **max\_file\_descriptors** 값을 확인하고 필요한 경우 구성합니다.

#### 추가 리소스

- [Elasticsearch 경고 규칙](#)에서 "ElasticsearchHighFileDescriptorUsage"를 검색합니다.
- [OpenShift Logging 대시보드](#)에서 "File Descriptors In Use"를 검색합니다.

## 13장. OPENSIFT LOGGING 설치 제거

OpenShift Container Platform 클러스터에서 OpenShift Logging을 제거할 수 있습니다.

### 13.1. OPENSIFT CONTAINER PLATFORM에서 OPENSIFT LOGGING 설치 삭제

**ClusterLogging** 사용자 정의 리소스(CR)를 삭제하여 로그 집계를 중지할 수 있습니다. CR을 삭제한 후에 다른 OpenShift Logging 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

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

OpenShift Logging을 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 인스턴스를 클릭합니다.

- d. 인스턴스 옆에 있는 옵션 메뉴  를 클릭하고 **ClusterLogging** 삭제를 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 삭제를 선택합니다.

- c. **ClusterLogging** 옆에 있는 옵션 메뉴  를 클릭하고 사용자 정의 리소스 정의 삭제를 선택합니다.

- d. **Elasticsearch** 옆에 있는 옵션 메뉴  를 클릭하고 사용자 정의 리소스 정의 삭제를 선택합니다.

3. 선택 사항: Red Hat OpenShift Logging Operator 및 OpenShift Elasticsearch Operator를 제거합니다.

- a. Operator → 설치된 Operator 페이지로 전환합니다.

b. Red Hat OpenShift Logging Operator 옆에 있는 옵션 메뉴  를 클릭하고 Operator 설치 제거를 선택합니다.

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

4. 선택 사항: OpenShift Logging 및 Elasticsearch 프로젝트를 제거합니다.

a. 홈 → 프로젝트 페이지로 전환합니다.

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

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

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

c. 스토리지 공간을 복구하려면 PV를 삭제할 수 있습니다.

- 수동으로 영구 불륨 회수

## 14장. 로그 레코드 필드

OpenShift Logging에서 내보낸 로그 레코드에 다음 필드가 표시될 수 있습니다. 로그 레코드는 일반적으로 JSON 개체로 포맷되지만 동일한 데이터 모델을 다른 인코딩에 적용할 수 있습니다.

Elasticsearch 및 Kibana에서 이러한 필드를 검색하려면 검색할 때 전체 점선 필드 이름을 사용합니다. 예를 들어 Elasticsearch /\_search URL로 Kubernetes Pod 이름을 찾으려면 `/_search/q=kubernetes.pod_name:name-of-my-pod`를 사용합니다.

최상위 수준 필드는 모든 레코드에 있을 수 있습니다.

## 15장. MESSAGE

원본 로그 항목 텍스트 UTF-8로 인코딩됩니다. 비어 있지 않은 **structured** 필드가 있는 경우 이 필드가 없거나 비어 있을 수 있습니다. **structured** 대한 자세한 내용은 설명을 참조하십시오.

데이터 유형	text
예시 값	<b>HAPPY</b>

## 16장. STRUCTURED

구조화된 오브젝트인 원본 로그 항목입니다. 이 필드는 Forwarder가 구조화된 JSON 로그를 구문 분석하도록 구성된 경우에 존재할 수 있습니다. 원본 로그 항목이 유효한 구조화된 로그인 경우 이 필드에는 동일한 JSON 구조가 포함됩니다. 그렇지 않으면 이 필드는 비어 있거나 없으며 **message** 필드에는 원래 로그 메시지가 포함됩니다. **structured** 필드에는 로그 메시지에 포함된 하위 필드가 있을 수 있으며 여기에 정의된 제한 사항이 없습니다.

데이터 유형	group
예시 값	map[message:starting fluentd worker pid=21631 ppid=21618 worker=0 pid:21631 ppid:21618 worker:0]

## 17장. @TIMESTAMP

로그 페이로드가 작성되거나 작성 시간을 알 수 없는 경우 로그 페이로드가 처음 수집될 때 표시되는 UTC 값입니다. "@" 접두사는 특정 용도로 예약된 필드를 나타냅니다. 대부분의 도구가 기본적으로 Elasticsearch를 사용하여 "@timestamp"를 찾습니다.

데이터 유형	date
예시 값	<b>2015-01-24 14:06:05.071000000 Z</b>

## 18장. 호스트 이름

이 로그 메시지가 시작된 호스트의 이름입니다. Kubernetes 클러스터에서 이는 **kubernetes.host**와 동일합니다.

데이터 유형	keyword
--------	---------

## 19장. IPADDR4

소스 서버의 IPv4 주소입니다. 배열이 될 수 있습니다.

데이터 유형	ip
--------	----

## 20장. IPADDR6

사용 가능한 경우 소스 서버의 IPv6 주소입니다. 배열이 될 수 있습니다.

데이터 유형	ip
--------	----

## 21장. LEVEL

**rsyslog(severitytext property)** Python 로깅 모듈 등을 비롯한 다양한 소스의 로깅 수준입니다.

다음 값은 **syslog.h**에서 가져오고 그 앞에 해당하는 숫자가 옵니다.

- **0 = emerg**, 시스템을 사용할 수 없습니다.
- **1 = alert**, 즉시 조치를 취해야 합니다.
- **2 = crit**, 심각한 상태입니다.
- **3 = err**, 오류 상태입니다.
- **4 = warn**, 경고 상태입니다.
- **5 = notice**, 정상이지만 중요한 상태입니다.
- **6 = info**, 정보를 제공합니다.
- **7 = debug**, 디버그 수준 메시지입니다.

다음 두 값은 **syslog.h**의 일부가 아니지만 널리 사용됩니다.

- **8 = trace**, trace-level 메시지는 **debug** 메시지보다 더 자세합니다.
- **9 = unknown**, 로깅 시스템에서 인식하지 않는 값을 얻는 경우입니다.

다른 로깅 시스템의 로그 수준 또는 우선 순위를 이전 목록의 가장 가까운 일치 항목에 매핑합니다. 예를 들어 **python 로깅**에서는 **CRITICAL**은 **crit**로 **ERROR**는 **err** 등과 일치시킬 수 있습니다.

데이터 유형	keyword
예시 값	<b>info</b>

## 22장. PID

사용 가능한 경우 이는 로깅 엔터티의 프로세스 ID입니다.

데이터 유형	keyword
--------	---------

## 23장. SERVICE

사용 가능한 경우 로깅 엔티티와 연관된 서비스의 이름입니다. 예를 들어 syslog의 **APP-NAME** 및 rsyslog의 **programname** 속성은 서비스 필드에 매핑됩니다.

데이터 유형	keyword
--------	---------

## 24장. TAGS

선택 사항: 수집기 또는 정규화기에 의해 각 로그에 배치된 Operator 정의 태그 목록을 제공합니다. 페이로드는 공백으로 구분된 문자열 토큰이 있는 문자열이거나 문자열 토큰의 JSON 목록일 수 있습니다.

데이터 유형	text
--------	------

## 25장. FILE

수집기에서 이 로그 항목을 읽는 로그 파일의 경로입니다. 일반적으로 클러스터 노드의 **/var/log** 파일 시스템에 있는 경로입니다.

데이터 유형	text
--------	------

## 26장. OFFSET

오프셋 값입니다. 단일 로그 파일의 컨텍스트에서 값이 엄격하게 단조롭게 증가하는 한 파일에서 로그 라인의 시작까지의 바이트 수(0 또는 1기반) 또는 로그 라인 번호(0 또는 1기반)를 나타낼 수 있습니다. 새 버전의 로그 파일(회전)을 나타내는 값을 줄바꿈할 수 있습니다.

데이터 유형	long
--------	------

## 27장. KUBERNETES

쿠버네티스 관련 메타데이터의 네임스페이스입니다.

데이터 유형	group
--------	-------

### 27.1. KUBERNETES.POD\_NAME

Pod의 이름입니다.

데이터 유형	keyword
--------	---------

### 27.2. KUBERNETES.POD\_ID

Pod의 Kubernetes ID입니다.

데이터 유형	keyword
--------	---------

### 27.3. KUBERNETES.NAMESPACE\_NAME

Kubernetes의 네임스페이스 이름입니다.

데이터 유형	keyword
--------	---------

### 27.4. KUBERNETES.NAMESPACE\_ID

Kubernetes의 네임스페이스 ID입니다.

데이터 유형	keyword
--------	---------

### 27.5. KUBERNETES.HOST

Kubernetes 노드 이름입니다.

데이터 유형	keyword
--------	---------

### 27.6. KUBERNETES.CONTAINER\_NAME

Kubernetes의 컨테이너 이름입니다.

데이터 유형	keyword
--------	---------

## 27.7. KUBERNETES.ANNOTATIONS

Kubernetes 오브젝트와 관련된 주석입니다.

데이터 유형	group
--------	-------

## 27.8. KUBERNETES.LABELS

원래 Kubernetes Pod에 있는 레이블입니다.

데이터 유형	group
--------	-------

## 27.9. KUBERNETES.EVENT

Kubernetes 마스터 API에서 얻은 Kubernetes 이벤트입니다. 이 이벤트 설명은 [Event v1 코어](#)의 **type Event**를 대략적으로 따릅니다.

데이터 유형	group
--------	-------

### 27.9.1. kubernetes.event.verb

이벤트 유형, **ADDED**, **MODIFIED** 또는 **DELETED**

데이터 유형	keyword
예시 값	<b>ADDED</b>

### 27.9.2. kubernetes.event.metadata

이벤트 생성 위치 및 시간 관련 정보입니다.

데이터 유형	group
--------	-------

#### 27.9.2.1. kubernetes.event.metadata.name

이벤트 생성을 트리거한 오브젝트의 이름입니다.

데이터 유형	keyword
예시 값	<b>java-mainclass-1.14d888a4cfc24890</b>

#### 27.9.2.2. kubernetes.event.metadata.namespace

이벤트가 처음 발생한 네임스페이스의 이름입니다. **eventrouter** 애플리케이션이 배포된 네임스페이스인 **kubernetes.namespace\_name**과 다릅니다.

데이터 유형	keyword
예시 값	<b>default</b>

### 27.9.2.3. kubernetes.event.metadata.selfLink

이벤트에 대한 링크입니다.

데이터 유형	keyword
예시 값	<b>/api/v1/namespaces/javaj/events/java-mainclass-1.14d888a4cfc24890</b>

### 27.9.2.4. kubernetes.event.metadata.uid

이벤트의 고유 ID입니다.

데이터 유형	keyword
예시 값	<b>d828ac69-7b58-11e7-9cf5-5254002f560c</b>

### 27.9.2.5. kubernetes.event.metadata.resourceVersion

서버의 내부 버전의 이벤트를 식별하는 문자열입니다. 클라이언트는 이 문자열을 사용하여 오브젝트가 변경 될 시기를 결정할 수 있습니다.

데이터 유형	integer
예시 값	<b>311987</b>

## 27.9.3. kubernetes.event.involvedObject

이벤트의 오브젝트입니다.

데이터 유형	group
--------	-------

### 27.9.3.1. kubernetes.event.involvedObject.kind

오브젝트 유형입니다.

데이터 유형	keyword
예시 값	<b>ReplicationController</b>

### 27.9.3.2. kubernetes.event.involvedObject.namespace

관련 오브젝트의 네임스페이스 이름입니다. `eventrouter` 애플리케이션이 배포된 네임스페이스인 `kubernetes.namespace_name`과 다를 수 있습니다.

데이터 유형	keyword
예시 값	<b>default</b>

### 27.9.3.3. kubernetes.event.involvedObject.name

이벤트를 트리거한 오브젝트의 이름입니다.

데이터 유형	keyword
예시 값	<b>java-mainclass-1</b>

### 27.9.3.4. kubernetes.event.involvedObject.uid

오브젝트의 고유 ID입니다.

데이터 유형	keyword
예시 값	<b>e6bff941-76a8-11e7-8193-5254002f560c</b>

### 27.9.3.5. kubernetes.event.involvedObject.apiVersion

kubernetes 마스터 API의 버전입니다.

데이터 유형	keyword
예시 값	<b>v1</b>

### 27.9.3.6. kubernetes.event.involvedObject.resourceVersion

이벤트를 트리거한 서버의 내부 버전을 식별하는 문자열입니다. 클라이언트는 이 문자열을 사용하여 오브젝트가 변경될 시기를 결정할 수 있습니다.

데이터 유형	keyword
예시 값	<b>308882</b>

## 27.9.4. kubernetes.event.reason

이 이벤트를 생성하는 이유를 제공하는 짧은 머신 이해 문자열입니다.

데이터 유형	keyword
--------	---------

예시 값	<b>SuccessfulCreate</b>
------	-------------------------

### 27.9.5. kubernetes.event.source\_component

이 이벤트를 보고한 구성 요소입니다.

데이터 유형	keyword
예시 값	<b>replication-controller</b>

### 27.9.6. kubernetes.event.firstTimestamp

이벤트가 처음 기록된 시간입니다.

데이터 유형	date
예시 값	<b>2017-08-07 10:11:57.000000000 Z</b>

### 27.9.7. kubernetes.event.count

이 이벤트가 발생한 횟수입니다.

데이터 유형	integer
예시 값	<b>1</b>

### 27.9.8. kubernetes.event.type

이벤트 유형, **Normal** 또는 **Warning**입니다. 새 유형을 나중에 추가할 수 있습니다.

데이터 유형	keyword
예시 값	<b>Normal</b>

## 28장. OPENSIFT

openshift-logging 특정 메타데이터의 네임스페이스

데이터 유형	group
--------	-------

### 28.1. OPENSIFT.LABELS

Cluster Log Forwarder 구성에 추가된 레이블

데이터 유형	group
--------	-------