



OpenShift Container Platform 4.12

로깅

OpenShift Logging 설치, 사용법, 릴리스 정보

OpenShift Container Platform 4.12 로깅

OpenShift Logging 설치, 사용법, 릴리스 정보

Legal Notice

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

Node.js ® 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 ® 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.

Abstract

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

Table of Contents

1장. 릴리스 노트	5
1.1. LOGGING 5.8	5
1.2. LOGGING 5.7	15
1.3. 로깅 5.6	33
1.4. LOGGING 5.5	48
2장. 지원	71
2.1. 지원되는 API 사용자 정의 리소스 정의	72
2.2. 지원되지 않는 로깅 구성	72
2.3. 관리되지 않는 OPERATOR에 대한 지원 정책	73
2.4. RED HAT 지원을 위한 로깅 데이터 수집	74
3장. 로깅 문제 해결	77
3.1. 로깅 상태 보기	77
3.2. 로그 전달 문제 해결	84
3.3. 로깅 경고 문제 해결	87
3.4. ELASTICSEARCH 로그 저장소의 상태 보기	101
4장. 로깅 정보	114
4.1. 로깅 아키텍처	114
4.2. 로깅 배포 정보	116
5장. 로깅 설치	120
5.1. 웹 콘솔을 사용하여 RED HAT OPENSHIFT LOGGING OPERATOR 설치	120
5.2. 웹 콘솔을 사용하여 CLUSTERLOGGING 오브젝트 생성	122
5.3. CLI를 사용하여 RED HAT OPENSHIFT LOGGING OPERATOR 설치	124
5.4. CLI를 사용하여 CLUSTERLOGGING 오브젝트 생성	127
5.5. 설치 후 작업	131
6장. 로깅 업데이트	140
6.1. 마이너 릴리스 업데이트	140
6.2. 주요 릴리스 업데이트	140
6.3. 모든 네임스페이스를 조사하도록 RED HAT OPENSHIFT LOGGING OPERATOR 업그레이드	140
6.4. RED HAT OPENSHIFT LOGGING OPERATOR 업데이트	142
6.5. LOKI OPERATOR 업데이트	143
6.6. OPENSHIFT ELASTICSEARCH OPERATOR 업데이트	144
7장. 로그 시각화	150
7.1. 로그 시각화 정보	150
7.2. 웹 콘솔을 사용한 로그 시각화	154
7.3. 클러스터 대시보드 보기	157
7.4. KIBANA를 사용한 로그 시각화	165
8장. 로깅 배포 구성	173
8.1. 로깅 구성 요소에 대한 CPU 및 메모리 제한 구성	173
8.2. SYSTEMD-JOURNALD 및 FLUENTD 구성	174
9장. 로그 수집 및 전달	179
9.1. 로그 수집 및 전달 정보	179
9.2. 로그 출력 유형	187
9.3. JSON 로그 전달 활성화	189
9.4. 로그 전달 구성	198
9.5. 로깅 수집기 구성	259

9.6. 쿠버네티스 이벤트 수집 및 저장	272
10장. 로그 스토리지	279
10.1. 로그 스토리지 정보	279
10.2. 로그 스토리지 설치	280
10.3. LOKISTACK 로그 저장소 구성	307
10.4. ELASTICSEARCH 로그 저장소 구성	323
11장. 로깅 경고	349
11.1. 기본 로깅 경고	349
12장. 성능 및 안정성 튜닝	353
12.1. 흐름 제어 메커니즘	353
13장. 리소스 예약	359
13.1. 노드 선택기를 사용하여 로깅 리소스 이동	359
13.2. 테인트 및 허용 오차를 사용하여 로깅 POD 배치 제어	372
14장. 로깅 설치 제거	386
14.1. 로깅 설치 제거	386
14.2. 로깅 PVC 삭제	388
14.3. LOKI 설치 제거	388
14.4. ELASTICSEARCH 설치 제거	390
14.5. CLI를 사용하여 클러스터에서 OPERATOR 삭제	392
15장. 로그 레코드 펠드	394
MESSAGE	394
STRUCTURED	394
@TIMESTAMP	394
호스트 이름	395
IPADDR4	395
IPADDR6	395
LEVEL	395
PID	396
SERVICE	396
16장. TAGS	398
FILE	398
OFFSET	398
17장. KUBERNETES	399
17.1. KUBERNETES.POD_NAME	399
17.2. KUBERNETES.POD_ID	399
17.3. KUBERNETES.NAMESPACE_NAME	399
17.4. KUBERNETES.NAMESPACE_ID	399
17.5. KUBERNETES.HOST	399
17.6. KUBERNETES.CONTAINER_NAME	400
17.7. KUBERNETES.ANNOTATIONS	400
17.8. KUBERNETES.LABELS	400
17.9. KUBERNETES.EVENT	400
18장. OPENSHIFT	406
18.1. OPENSHIFT.LABELS	406
19장. API 참조	407
19.1. 5.6 로깅 API 참조	407

20장. 용어집	452
----------------	-----

1장. 릴리스 노트

1.1. LOGGING 5.8



참고

로깅은 핵심 OpenShift Container Platform과 별도의 릴리스 주기와 함께 설치 가능한 구성 요소로 제공됩니다. [Red Hat OpenShift Container Platform 라이프 사이클 정책](#)은 릴리스 호환성에 대해 간단히 설명합니다.



참고

stable 채널은 최신 로깅 릴리스에 대한 업데이트만 제공합니다. 이전 릴리스에 대한 업데이트를 계속 받으려면 서브스크립션 채널을 **stable-x.y**로 변경해야 합니다. 여기서 **x.y**는 설치한 로깅 및 마이너 버전을 나타냅니다. 예를 들면 **stable-5.7**입니다.

1.1.1. Logging 5.8.5

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.8.5](#) 가 포함되어 있습니다.

1.1.1.1. 버그 수정

- 이번 업데이트 이전에는 Loki Operator의 **ServiceMonitor** 구성이 여러 Kubernetes 서비스와 일치하여 Loki Operator의 메트릭이 여러 번 수집될 수 있었습니다. 이번 업데이트를 통해 **ServiceMonitor**의 구성은 이제 전용 메트릭 서비스와만 일치합니다. ([LOG-5250](#))
- 이번 업데이트 이전에는 Red Hat 빌드 파이프라인에서 Loki 빌드의 기존 빌드 세부 정보와 revision, 분기, 버전과 같은 생략된 정보를 사용하지 않았습니다. 이번 업데이트를 통해 Red Hat 빌드 파이프라인에서 Loki 빌드에 이러한 세부 정보를 추가하여 문제를 해결합니다. ([LOG-5201](#))
- 이번 업데이트 이전에는 Loki Operator에서 **LokiStack**이 준비되었는지 확인하기 위해 Pod가 실행 중인지 확인했습니다. 이번 업데이트를 통해 **LokiStack**의 준비 상태가 해당 구성 요소의 상태를 반영하도록 Pod가 준비되었는지도 확인합니다. ([LOG-5171](#))
- 이번 업데이트 이전에는 로그 메트릭에 대한 쿼리를 실행하면 히스토그램에 오류가 발생했습니다. 이번 업데이트를 통해 히스토그램 토글 기능 및 히스토그램이 로그 메트릭에서 작동하지 않기 때문에 히스토그램 토글 기능이 비활성화되고 숨겨집니다. ([LOG-5044](#))
- 이번 업데이트 이전에는 Loki 및 Elasticsearch 번들에 잘못된 **maxOpenShiftVersion**이 있어 **IncompatibleOperatorsInstalled** 경고가 생성되었습니다. 번들의 4.16을 **maxOpenShiftVersion** 속성으로 포함하여 이 업데이트를 통해 문제가 해결되었습니다. ([LOG-5272](#))
- 이번 업데이트 이전에는 빌드 파이프라인에 빌드 날짜에 대한 링커 플래그가 포함되어 있지 않아 Loki 빌드에 **buildDate** 및 **goVersion**의 빈 문자열이 표시되었습니다. 이번 업데이트를 통해 빌드 파이프라인에 누락된 링커 플래그를 추가하면 문제가 해결되었습니다. ([LOG-5274](#))
- 이번 업데이트 이전에는 LogQL 구문 분석의 버그로 인해 쿼리에서 일부 줄 필터가 남겼습니다. 이번 업데이트를 통해 이제 구문 분석에 원래 쿼리를 변경하지 않고 모든 줄 필터가 포함됩니다. ([LOG-5270](#))
- 이번 업데이트 이전에는 **openshift-operators-redhat** 네임스페이스의 Loki Operator **ServiceMonitor**에서 인증에 정적 토큰과 CA 파일을 사용하여 **ServiceMonitor** 구성의 User

Workload Monitoring 사양의 Prometheus Operator에 오류가 발생했습니다. 이번 업데이트를 통해 **openshift-operators-redhat** 네임스페이스의 Loki Operator **ServiceMonitor**는 이제 **LocalReference** 오브젝트의 서비스 계정 토큰 시크릿을 참조합니다. 이 방법을 사용하면 Prometheus Operator의 User Workload Monitoring 사양에서 Loki Operator **ServiceMonitor**를 성공적으로 처리할 수 있으므로 Prometheus가 Loki Operator 메트릭을 스크랩할 수 있습니다. ([LOG-5240](#))

1.1.1.2. CVE

- [CVE-2023-5363](#)
- [CVE-2023-5981](#)
- [CVE-2023-6135](#)
- [CVE-2023-46218](#)
- [CVE-2023-48795](#)
- [CVE-2023-51385](#)
- [CVE-2024-0553](#)
- [CVE-2024-0567](#)
- [CVE-2024-24786](#)
- [CVE-2024-28849](#)

1.1.2. Logging 5.8.4

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.8.4 가 포함되어 있습니다.

1.1.2.1. 버그 수정

- 이번 업데이트 이전에는 개발자 콘솔의 로그에서 현재 네임스페이스를 고려하지 않아 클러스터 전체 로그 액세스 권한이 없는 사용자에게 쿼리 거부가 발생했습니다. 이번 업데이트를 통해 지원되는 모든 OCP 버전은 올바른 네임스페이스를 포함해야 합니다. ([LOG-4905](#))
- 이번 업데이트 이전에는 Cluster Logging Operator가 기본 로그 출력이 LokiStack인 경우에만 LokiStack 배포를 지원하는 **ClusterRoles**를 배포했습니다. 이번 업데이트를 통해 역할은 읽기 및 쓰기의 두 그룹으로 나뉩니다. 쓰기 역할은 이전에 수행하는 데 사용되는 모든 역할과 마찬가지로 기본 로그 스토리지 설정을 기반으로 배포됩니다. 읽기 역할은 로깅 콘솔 플러그인이 활성화되어 있는지 여부에 따라 배포됩니다. ([LOG-4987](#))
- 이번 업데이트 이전에는 하나의 서비스에서 **ownerReferences**를 변경하기 때문에 동일한 입력 수신자 이름을 정의하는 **ClusterLogForwarders**가 서비스를 끝없이 조정했습니다. 이번 업데이트를 통해 각 수신자 입력은 <CLF.Name>-<input.Name> 규칙을 사용하여 이름이 지정된 자체 서비스를 갖습니다. ([LOG-5009](#))
- 이번 업데이트 이전에는 시크릿 없이 cloudwatch로 로그를 전달할 때 **ClusterLogForwarder**에서 오류를 보고하지 않았습니다. 이번 업데이트를 통해 시크릿 없이 cloudwatch로 로그를 전달할 때 다음 오류 메시지가 표시됩니다. **cloudwatch** 출력에 시크릿을 제공해야 합니다. ([LOG-5021](#))

- 이번 업데이트 이전에는 **log_forwarder_input_info**에 애플리케이션, 인프라 및 감사 입력 메트릭 포인트가 포함되었습니다. 이번 업데이트를 통해 **http**도 지표 지점으로 추가됩니다. ([LOG-5043](#))

1.1.2.2. CVE

- [CVE-2021-35937](#)
- [CVE-2021-35938](#)
- [CVE-2021-35939](#)
- [CVE-2022-3545](#)
- [CVE-2022-24963](#)
- [CVE-2022-36402](#)
- [CVE-2022-41858](#)
- [CVE-2023-2166](#)
- [CVE-2023-2176](#)
- [CVE-2023-3777](#)
- [CVE-2023-3812](#)
- [CVE-2023-4015](#)
- [CVE-2023-4622](#)
- [CVE-2023-4623](#)
- [CVE-2023-5178](#)
- [CVE-2023-5363](#)
- [CVE-2023-5388](#)
- [CVE-2023-5633](#)
- [CVE-2023-6679](#)
- [CVE-2023-7104](#)
- [CVE-2023-27043](#)
- [CVE-2023-38409](#)
- [CVE-2023-40283](#)
- [CVE-2023-42753](#)
- [CVE-2023-43804](#)
- [CVE-2023-45803](#)

- CVE-2023-46813
- CVE-2024-20918
- CVE-2024-20919
- CVE-2024-20921
- CVE-2024-20926
- CVE-2024-20945
- CVE-2024-20952

1.1.3. Logging 5.8.3

이 릴리스에는 로깅 버그 수정 5.8.3 및 로깅 보안 수정 5.8.3이 포함되어 있습니다.

1.1.3.1. 버그 수정

- 이번 업데이트 이전에는 사용자 정의 S3 인증 기관을 읽도록 구성된 경우 Loki Operator는 ConfigMap 이름 또는 콘텐츠가 변경된 경우 구성을 자동으로 업데이트하지 않습니다. 이번 업데이트를 통해 Loki Operator에서 ConfigMap에 대한 변경 사항을 조사하고 생성된 구성을 자동으로 업데이트합니다. ([LOG-4969](#))
- 이번 업데이트 이전에는 유효한 URL 없이 Loki 출력이 구성되어 수집기 Pod가 충돌했습니다. 이번 업데이트를 통해 출력에 URL 검증이 적용되어 문제를 해결합니다. ([LOG-4822](#))
- 이번 업데이트 이전에는 Cluster Logging Operator가 서비스 계정 전달자 토큰을 사용할 시크릿을 지정하지 않은 출력에 대한 수집기 구성 필드를 생성합니다. 이번 업데이트를 통해 출력에 인증이 필요하지 않아 문제를 해결합니다. ([LOG-4962](#))
- 이번 업데이트 이전에는 출력의 **tls.insecureSkipVerify** 필드가 보안을 정의하지 않고 **true** 값으로 설정되지 않았습니다. 이번 업데이트를 통해 이 값을 설정하는 데 더 이상 시크릿이 필요하지 않습니다. ([LOG-4963](#))
- 이번 업데이트 이전에는 출력 구성에서 비보안 (HTTP) URL과 TLS 인증의 조합을 허용했습니다. 이번 업데이트를 통해 TLS 인증을 위해 구성된 출력에 보안(HTTPS) URL이 필요합니다. ([LOG-4893](#))

1.1.3.2. CVE

- CVE-2021-35937
- CVE-2021-35938
- CVE-2021-35939
- CVE-2023-7104
- CVE-2023-27043
- CVE-2023-48795
- CVE-2023-51385

- CVE-2024-0553

1.1.4. Logging 5.8.2

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.8.2](#) 가 포함되어 있습니다.

1.1.4.1. 버그 수정

- 이번 업데이트 이전에는 LokiStack 규칙의 Pod에서 교차 Pod 통신에 사용되는 HTTP URL의 IPv6 Pod IP를 포맷하지 않아 Prometheus 호환 API를 통한 규칙 및 경고를 쿼리할 수 없었습니다. 이번 업데이트를 통해 LokiStack 룰러 Pod는 IPv6 포드 IP를 대괄호로 캡슐화하여 문제를 해결합니다. ([LOG-4890](#))
- 이번 업데이트 이전에는 개발자 콘솔 로그에서 현재 네임스페이스를 고려하지 않아 클러스터 전체 로그 액세스 권한이 없는 사용자에게 쿼리 거부가 발생했습니다. 이번 업데이트를 통해 네임스페이스 포함이 수정되어 문제를 해결합니다. ([LOG-4947](#))
- 이번 업데이트 이전에는 OpenShift Container Platform 웹 콘솔의 로깅 보기 플러그인에서 사용자 정의 노드 배치 및 허용 오차를 허용하지 않았습니다. 이번 업데이트를 통해 OpenShift Container Platform 웹 콘솔의 로깅 보기 플러그인에 사용자 정의 노드 배치 및 허용 오차가 추가되었습니다. ([LOG-4912](#))

1.1.4.2. CVE

- CVE-2022-44638
- CVE-2023-1192
- CVE-2023-5345
- CVE-2023-20569
- CVE-2023-26159
- CVE-2023-39615
- CVE-2023-45871

1.1.5. Logging 5.8.1

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.8.1](#) 및 [OpenShift Logging 버그 수정 릴리스 5.8.1 Kibana](#) 가 포함되어 있습니다.

1.1.5.1. 기능 개선

1.1.5.1.1. 로그 컬렉션

- 이번 업데이트를 통해 Vector를 수집기로 구성하는 동안 Red Hat OpenShift Logging Operator에 논리를 추가하여 서비스 계정과 연결된 토큰 대신 시크릿에 지정된 토큰을 사용할 수 있습니다. ([LOG-4780](#))
- 이번 업데이트를 통해 **BoltDB shipper Loki** 대시보드의 이름이 **Index** 대시보드로 변경되었습니다. ([LOG-4828](#))

1.1.5.2. 버그 수정

- 이번 업데이트 이전에는 롤오버 조건이 충족되지 않은 경우에도 **ClusterLogForwarder**에서 JSON 로그의 구문 분석을 활성화한 후 빈 인덱스를 생성했습니다. 이번 업데이트를 통해 **write-index** 가 비어 있을 때 **ClusterLogForwarder**에서 롤오버를 건너뜁니다. ([LOG-4452](#))
- 이번 업데이트 이전에는 백터가 기본 로그 수준을 잘못 설정했습니다. 이번 업데이트를 통해 로그 수준 탐지를 위해 정규식 또는 **regexp** 의 향상된 기능을 개선하여 올바른 로그 수준이 설정됩니다. ([LOG-4480](#))
- 이번 업데이트 이전에는 인덱스 패턴을 생성하는 프로세스 중에 각 로그 출력의 초기 인덱스에서 기본 별칭이 누락되었습니다. 결과적으로 Kibana 사용자는 OpenShift Elasticsearch Operator를 사용하여 인덱스 패턴을 생성할 수 없었습니다. 이번 업데이트에서는 OpenShift Elasticsearch Operator에 누락된 별칭이 추가되어 문제를 해결합니다. Kibana 사용자는 **{app,infra,audit}-000001** 인덱스를 포함하는 인덱스 패턴을 생성할 수 있습니다. ([LOG-4683](#))
- 이번 업데이트 이전에는 IPv6 클러스터에서 Prometheus 서버가 바인딩되어 Fluentd 수집기 Pod가 **CrashLoopBackOff** 상태에 있었습니다. 이번 업데이트를 통해 수집기는 IPv6 클러스터에서 제대로 작동합니다. ([LOG-4706](#))
- 이번 업데이트 이전에는 **ClusterLogForwarder** 가 변경될 때마다 Red Hat OpenShift Logging Operator가 다양한 조정을 수행했습니다. 이번 업데이트를 통해 Red Hat OpenShift Logging Operator는 조정을 트리거한 수집기 데몬 세트의 상태 변경을 무시합니다. ([LOG-4741](#))
- 이번 업데이트 이전에는 **{ibm-power-title}** 머신의 Vector 로그 수집기 Pod가 **CrashLoopBackOff** 상태로 중단되었습니다. 이번 업데이트를 통해 Vector 로그 수집기 Pod가 **{ibm-power-title}** 아키텍처 시스템에서 성공적으로 시작됩니다. ([LOG-4768](#))
- 이번 업데이트 이전에는 내부 LokiStack에 대한 레거시 전달자를 사용하여 전달하면 Fluentd 수집기 Pod를 사용하여 SSL 인증서 오류가 발생했습니다. 이번 업데이트를 통해 로그 수집기 서비스 계정은 기본적으로 연결된 토큰과 **ca.crt** 를 사용하여 인증에 사용됩니다. ([LOG-4791](#))
- 이번 업데이트 이전에는 내부 LokiStack에 대한 레거시 전달자를 사용하여 전달하면 Vector 수집기 Pod를 사용하여 SSL 인증서 오류가 발생했습니다. 이번 업데이트를 통해 기본적으로 로그 수집기 서비스 계정이 인증에 사용되며 연결된 토큰 및 **ca.crt** 를 사용합니다. ([LOG-4852](#))
- 이번 수정 이전에는 자리 표시자에 대해 호스트 또는 여러 호스트를 평가한 후 IPv6 주소가 올바르게 구문 분석되지 않았습니다. 이번 업데이트를 통해 IPv6 주소가 올바르게 구문 분석됩니다. ([LOG-4811](#))
- 이번 업데이트 이전에는 HTTP 수신자 입력에 대한 감사 권한을 수집하기 위해 **ClusterRoleBinding** 을 생성해야 했습니다. 이번 업데이트를 통해 끝점이 이미 클러스터 인증기관에 따라 다르기 때문에 **ClusterRoleBinding** 을 생성할 필요가 없습니다. ([LOG-4815](#))
- 이번 업데이트 이전에는 Loki Operator에서 사용자 정의 CA 번들을 룰러 Pod에 마운트하지 않았습니다. 결과적으로 경고 또는 레코딩 규칙을 평가하는 프로세스 중에 오브젝트 스토리지 액세스가 실패했습니다. 이번 업데이트를 통해 Loki Operator는 모든 룰러 Pod에 사용자 정의 CA 번들을 마운트합니다. 규칙자 Pod는 오브젝트 스토리지에서 로그를 다운로드하여 경고 또는 레코딩 규칙을 평가할 수 있습니다. ([LOG-4836](#))
- 이번 업데이트 이전에는 **ClusterLogForwarder**에서 **inputs.receiver** 섹션을 제거하는 동안 HTTP 입력 서비스 및 관련 보안이 삭제되지 않았습니다. 이번 업데이트를 통해 필요하지 않은 경우 HTTP 입력 리소스가 삭제됩니다. ([LOG-4612](#))
- 이번 업데이트 이전에는 **ClusterLogForwarder**에서 상태에 검증 오류가 표시되었지만 출력 및 파이프라인 상태가 특정 문제를 정확하게 반영하지 않았습니다. 이번 업데이트를 통해 출력, 입력

또는 필터가 잘못 구성된 경우 파이프라인 상태에 검증 실패 이유가 올바르게 표시됩니다. ([LOG-4821](#))

- 이번 업데이트 이전에는 시간 범위 또는 심각도와 같은 제어를 사용하는 **LogQL** 쿼리를 변경하면 레이블 matcher Operator가 정규식처럼 정의되었습니다. 이번 업데이트를 통해 쿼리를 업데이트 할 때 정규식 Operator가 변경되지 않은 상태로 유지됩니다. ([LOG-4841](#))

1.1.5.3. CVE

- [CVE-2007-4559](#)
- [CVE-2021-3468](#)
- [CVE-2021-3502](#)
- [CVE-2021-3826](#)
- [CVE-2021-43618](#)
- [CVE-2022-3523](#)
- [CVE-2022-3565](#)
- [CVE-2022-3594](#)
- [CVE-2022-4285](#)
- [CVE-2022-38457](#)
- [CVE-2022-40133](#)
- [CVE-2022-40982](#)
- [CVE-2022-41862](#)
- [CVE-2022-42895](#)
- [CVE-2023-0597](#)
- [CVE-2023-1073](#)
- [CVE-2023-1074](#)
- [CVE-2023-1075](#)
- [CVE-2023-1076](#)
- [CVE-2023-1079](#)
- [CVE-2023-1206](#)
- [CVE-2023-1249](#)
- [CVE-2023-1252](#)
- [CVE-2023-1652](#)
- [CVE-2023-1855](#)

- CVE-2023-1981
- CVE-2023-1989
- CVE-2023-2731
- CVE-2023-3138
- CVE-2023-3141
- CVE-2023-3161
- CVE-2023-3212
- CVE-2023-3268
- CVE-2023-3316
- CVE-2023-3358
- CVE-2023-3576
- CVE-2023-3609
- CVE-2023-3772
- CVE-2023-3773
- CVE-2023-4016
- CVE-2023-4128
- CVE-2023-4155
- CVE-2023-4194
- CVE-2023-4206
- CVE-2023-4207
- CVE-2023-4208
- CVE-2023-4273
- CVE-2023-4641
- CVE-2023-22745
- CVE-2023-26545
- CVE-2023-26965
- CVE-2023-26966
- CVE-2023-27522
- CVE-2023-29491

- CVE-2023-29499
- CVE-2023-30456
- CVE-2023-31486
- CVE-2023-32324
- CVE-2023-32573
- CVE-2023-32611
- CVE-2023-32665
- CVE-2023-33203
- CVE-2023-33285
- CVE-2023-33951
- CVE-2023-33952
- CVE-2023-34241
- CVE-2023-34410
- CVE-2023-35825
- CVE-2023-36054
- CVE-2023-37369
- CVE-2023-38197
- CVE-2023-38545
- CVE-2023-38546
- CVE-2023-39191
- CVE-2023-39975
- CVE-2023-44487

1.1.6. Logging 5.8.0

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.8.0 및 OpenShift Logging 버그 수정 릴리스 5.8.0 Kibana 가 포함되어 있습니다.

1.1.6.1. 사용 중단 알림

로깅 5.8에서 Elasticsearch, Fluentd, Kibana는 더 이상 사용되지 않으며 Logging 6.0에서 제거될 예정이며 향후 OpenShift Container Platform 릴리스와 함께 제공될 것으로 예상됩니다. Red Hat은 현재 릴리스 라이프사이클 동안 중요한 CVE 버그 수정 및 이러한 구성 요소에 대한 지원을 제공하지만 이러한 구성 요소는 더 이상 기능 개선 사항을 받지 않습니다. Loki Operator에서 제공하는 Red Hat OpenShift Logging

Operator 및 LokiStack에서 제공하는 Vector 기반 수집기는 로그 수집 및 스토리지에 권장되는 Operator입니다. 모든 사용자가 Vector 및 Loki 로그 스택을 채택하도록 권장합니다. 이는 앞으로 개선될 스택이 되기 때문입니다.

1.1.6.2. 기능 개선

1.1.6.2.1. 로그 컬렉션

- 이번 업데이트를 통해 LogFileMetricExporter는 기본적으로 수집기와 함께 더 이상 배포되지 않습니다. 컨테이너를 실행하여 생성된 로그에서 메트릭을 생성하려면 **LogFileMetricExporter** CR(사용자 정의 리소스)을 수동으로 생성해야 합니다. **LogFileMetricExporter** CR을 생성하지 않으면 OpenShift Container Platform 웹 콘솔 대시보드에 **Produced Logs**의 **No datapoints found** 메시지가 표시될 수 있습니다. ([LOG-3819](#))
- 이번 업데이트를 통해 모든 네임스페이스에 여러 개의 격리된 **ClusterLogForwarder** CR(사용자 정의 리소스) 인스턴스를 배포할 수 있습니다. 이를 통해 독립 그룹은 다른 컬렉터 배포에서 구성 을 격리하는 동안 원하는 로그를 대상으로 전달할 수 있습니다. ([LOG-1343](#))



중요

openshift-logging 네임스페이스 이외의 추가 네임스페이스에서 다중 클러스터 로그 전달을 지원하려면 모든 네임스페이스를 조사하도록 Red Hat OpenShift Logging Operator를 업데이트해야 합니다. 이 기능은 새로운 Red Hat OpenShift Logging Operator 버전 5.8 설치에서 기본적으로 지원됩니다.

- 이번 업데이트를 통해 흐름 제어 또는 속도 제한 메커니즘을 사용하여 과도한 로그 레코드를 삭제 하여 수집하거나 전달할 수 있는 로그 데이터의 볼륨을 제한할 수 있습니다. 입력 제한으로 인해 성능이 좋지 않은 컨테이너가 로깅을 과부하하는 것을 방지하고 출력 제한으로 인해 지정된 데이터 저장소에 전달되는 로그의 비율에 문제가 발생합니다. ([LOG-884](#))
- 이번 업데이트를 통해 HTTP 연결을 찾고 웹 후크라고도 하는 HTTP 서버로 로그를 수신하도록 로그 수집기를 구성할 수 있습니다. ([LOG-4562](#))
- 이번 업데이트를 통해 감사 정책을 구성하여 로그 수집기에서 전달하는 Kubernetes 및 OpenShift API 서버 이벤트를 제어 할 수 있습니다. ([LOG-3982](#))

1.1.6.2.2. 로그 스토리지

- 이번 업데이트를 통해 LokiStack 관리자는 네임스페이스에 따라 로그에 대한 액세스 권한을 부여 하여 누가 로그에 액세스할 수 있는 사용자를 보다 세밀하게 제어 할 수 있습니다. ([LOG-3841](#))
- 이번 업데이트를 통해 Loki Operator는 LokiStack 배포에 **PodDisruptionBudget** 구성을 도입하여 ingestion 및 쿼리 경로를 계속 사용하여 OpenShift Container Platform 클러스터를 재시작하는 동안 정상적인 작업을 보장합니다. ([LOG-3839](#))
- 이번 업데이트를 통해 기본 Affinity 및 anti-Affinity 정책 세트를 적용하여 기존 LokiStack 설치의 안정성이 원활하게 향상됩니다. ([LOG-3840](#))
- 이번 업데이트를 통해 영역 장애 발생 시 안정성을 높이기 위해 LokiStack에서 영역 인식 데이터 복제를 관리자로 관리 할 수 있습니다. ([LOG-3266](#))
- 이번 업데이트를 통해 몇 가지 워크로드와 작은 수집 볼륨(최대 100GB/일)을 호스팅하는 OpenShift Container Platform 클러스터에 지원되는 새로운 소규모 LokiStack 크기가 1x.extra-tekton가 도입되었습니다. ([LOG-4329](#))

- 이번 업데이트를 통해 LokiStack 관리자는 공식 Loki 대시보드에 액세스하여 스토리지 성능 및 각 구성 요소의 상태를 검사할 수 있습니다. ([LOG-4327](#))

1.1.6.2.3. 로그 콘솔

- 이번 업데이트를 통해 Elasticsearch가 기본 로그 저장소인 경우 로깅 콘솔 플러그인을 활성화할 수 있습니다. ([LOG-3856](#))
- 이번 업데이트를 통해 OpenShift Container Platform 애플리케이션 소유자는 OpenShift Container Platform 버전 4.14 이상의 OpenShift Container Platform 웹 콘솔 개발자 화면에서 애플리케이션 로그 기반 경고에 대한 알림을 수신할 수 있습니다. ([LOG-3548](#))

1.1.6.3. 확인된 문제

- 현재 Red Hat OpenShift Logging Operator의 버전 5.8로 업그레이드한 후 Splunk 로그 전달이 작동하지 않을 수 있습니다. 이 문제는 OpenSSL 버전 1.1.1에서 버전 3.0.7로 전환하여 발생합니다. 최신 OpenSSL 버전에서는 기본 동작 변경 사항이 있습니다. 여기서 [RFC 5746](#) 확장을 노출하지 않으면 TLS 1.2 끝점에 대한 연결이 거부됩니다. 이 문제를 해결하려면 Splunk HEC(HTTP 이벤트 수집기) 끝점 앞에 있는 TLS 종료 로드 밸런서에서 TLS 1.3 지원을 활성화합니다. Splunk는 타사 시스템이며 Splunk 끝에서 구성해야 합니다.
- 현재 HTTP/2 프로토콜에서 멀티플렉싱된 스트림을 처리하는 데 결함이 있습니다. 여기서 새로운 멀티플렉션을 반복적으로 요청하고 이를 취소하기 위해 **RST_STREAM** 프레임을 즉시 보낼 수 있습니다. 이로 인해 서버가 스트림을 설정하고 축소하여 서버 리소스 사용으로 인해 서비스가 거부되었습니다. 현재 이 문제에 대한 해결방법이 없습니다. ([LOG-4609](#))
- 현재 FluentD를 수집기로 사용할 때 수집기 Pod는 OpenShift Container Platform IPv6 지원 클러스터에서 시작할 수 없습니다. Pod 로그는 **fluentd pod [error]: unexpected error_class=SocketError error="getaddrinfo: Name 또는 service notknown 오류를 생성합니다.** 현재 이 문제에 대한 해결방법이 없습니다. ([LOG-4706](#))
- 현재 IPv6 지원 클러스터에서는 로그 경고를 사용할 수 없습니다. 현재 이 문제에 대한 해결방법이 없습니다. ([LOG-4709](#))
- 현재 **cluster-logging-rhel9-operator**에서 필요한 OpenSSL 라이브러리를 사용할 수 없기 때문에 FIPS 지원 클러스터에서 **must-gather**는 로그를 수집할 수 없습니다. 현재 이 문제에 대한 해결방법이 없습니다. ([LOG-4403](#))
- 현재 FIPS 지원 클러스터에 로깅 버전 5.8을 배포할 때 컬렉터 Pod는 시작할 수 없으며 FluentD를 수집기로 사용하는 동안 **CrashLoopBackOff** 상태에서 멈춥니다. 현재 이 문제에 대한 해결방법이 없습니다. ([LOG-3933](#))

1.1.6.4. CVE

- [CVE-2023-40217](#)

1.2. LOGGING 5.7



참고

로깅은 핵심 OpenShift Container Platform과 별도의 릴리스 주기와 함께 설치 가능한 구성 요소로 제공됩니다. [Red Hat OpenShift Container Platform 라이프 사이클 정책](#)은 릴리스 호환성에 대해 간단히 설명합니다.



참고

stable 채널은 최신 로깅 릴리스에 대한 업데이트만 제공합니다. 이전 릴리스에 대한 업데이트를 계속 받으려면 서브스크립션 채널을 **stable-x.y**로 변경해야 합니다. 여기서 **x.y**는 설치한 로깅 및 마이너 버전을 나타냅니다. 예를 들면 **stable-5.7**입니다.

1.2.1. Logging 5.7.12

이 릴리스에는 [OpenShift Logging 버그 수정 5.7.12](#) 가 포함되어 있습니다.

1.2.1.1. 버그 수정

- 이번 업데이트 이전에는 Loki Operator에서 **LokiStack**이 준비되었는지 확인하기 위해 Pod가 실행 중인지 확인했습니다. 이번 업데이트를 통해 **LokiStack**의 준비 상태가 해당 구성 요소의 상태를 반영하도록 Pod가 준비되었는지도 확인합니다. ([LOG-5172](#))
- 이번 업데이트 이전에는 Red Hat 빌드 파이프라인에서 Loki 빌드의 기존 빌드 세부 정보와 revision, 분기, 버전과 같은 생략된 정보를 사용하지 않았습니다. 이번 업데이트를 통해 Red Hat 빌드 파이프라인에서 Loki 빌드에 이러한 세부 정보를 추가하여 문제를 해결합니다. ([LOG-5202](#))
- 이번 업데이트 이전에는 Loki Operator의 **ServiceMonitor** 구성이 여러 Kubernetes 서비스와 일치하여 Loki Operator의 메트릭이 여러 번 수집될 수 있었습니다. 이번 업데이트를 통해 **ServiceMonitor**의 구성은 이제 전용 메트릭 서비스와만 일치합니다. ([LOG-5251](#))
- 이번 업데이트 이전에는 빌드 파이프라인에 빌드 날짜에 대한 링커 플래그가 포함되어 있지 않아 Loki 빌드에 **buildDate** 및 **goVersion**의 빈 문자열이 표시되었습니다. 이번 업데이트를 통해 빌드 파이프라인에 누락된 링커 플래그를 추가하면 문제가 해결되었습니다. ([LOG-5275](#))
- 이번 업데이트 이전에는 **openshift-operators-redhat** 네임스페이스의 Loki Operator **ServiceMonitor**에서 인증에 정적 토큰과 CA 파일을 사용하여 **ServiceMonitor** 구성의 User Workload Monitoring 사양의 Prometheus Operator에 오류가 발생했습니다. 이번 업데이트를 통해 **openshift-operators-redhat** 네임스페이스의 Loki Operator **ServiceMonitor**는 이제 **LocalReference** 오브젝트의 서비스 계정 토큰 시크릿을 참조합니다. 이 방법을 사용하면 Prometheus Operator의 User Workload Monitoring 사양에서 Loki Operator **ServiceMonitor**를 성공적으로 처리할 수 있으므로 Prometheus가 Loki Operator 메트릭을 스크랩할 수 있습니다. ([LOG-5241](#))

1.2.1.2. CVE

- [CVE-2021-35937](#)
- [CVE-2021-35938](#)
- [CVE-2021-35939](#)
- [CVE-2022-3545](#)
- [CVE-2022-41858](#)
- [CVE-2023-1073](#)
- [CVE-2023-1838](#)
- [CVE-2023-2166](#)

- CVE-2023-2176
- CVE-2023-4623
- CVE-2023-4921
- CVE-2023-5717
- CVE-2023-6135
- CVE-2023-6356
- CVE-2023-6535
- CVE-2023-6536
- CVE-2023-6606
- CVE-2023-6610
- CVE-2023-6817
- CVE-2023-7104
- CVE-2023-27043
- CVE-2023-40283
- CVE-2023-45871
- CVE-2023-46813
- CVE-2023-48795
- CVE-2023-51385
- CVE-2024-0553
- CVE-2024-0646
- CVE-2024-24786

1.2.2. 로깅 5.7.11

이 릴리스에는 [로깅 버그 수정 5.7.11](#) 이 포함되어 있습니다.

1.2.2.1. 버그 수정

- 이번 업데이트 이전에는 사용자 정의 S3 인증 기관을 읽도록 구성된 경우 Loki Operator는 **ConfigMap** 오브젝트의 이름 또는 콘텐츠가 변경된 경우 구성을 자동으로 업데이트하지 않았습니다. 이번 업데이트를 통해 Loki Operator는 **ConfigMap** 오브젝트의 변경 사항을 감시하고 생성된 구성을 자동으로 업데이트합니다. ([LOG-4968](#))

1.2.2.2. CVE

- CVE-2023-39326

1.2.3. Logging 5.7.10

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.7.10 이 포함되어 있습니다.

1.2.3.1. 버그 수정

이번 업데이트 이전에는 LokiStack 규칙 러 Pod에서 교차 Pod 통신에 사용되는 HTTP URL의 IPv6 Pod IP를 포맷하지 않아 Prometheus 호환 API를 통한 규칙 및 경고를 쿼리할 수 없었습니다. 이번 업데이트를 통해 LokiStack 런너 Pod는 IPv6 포드 IP를 대괄호로 캡슐화하여 문제를 해결합니다. ([LOG-4891](#))

1.2.3.2. CVE

- [CVE-2007-4559](#)
- [CVE-2021-43975](#)
- [CVE-2022-3594](#)
- [CVE-2022-3640](#)
- [CVE-2022-4285](#)
- [CVE-2022-4744](#)
- [CVE-2022-28388](#)
- [CVE-2022-38457](#)
- [CVE-2022-40133](#)
- [CVE-2022-40982](#)
- [CVE-2022-41862](#)
- [CVE-2022-42895](#)
- [CVE-2022-45869](#)
- [CVE-2022-45887](#)
- [CVE-2022-48337](#)
- [CVE-2022-48339](#)
- [CVE-2023-0458](#)
- [CVE-2023-0590](#)
- [CVE-2023-0597](#)
- [CVE-2023-1073](#)
- [CVE-2023-1074](#)
- [CVE-2023-1075](#)
- [CVE-2023-1079](#)

- CVE-2023-1118
- CVE-2023-1206
- CVE-2023-1252
- CVE-2023-1382
- CVE-2023-1855
- CVE-2023-1989
- CVE-2023-1998
- CVE-2023-2513
- CVE-2023-3138
- CVE-2023-3141
- CVE-2023-3161
- CVE-2023-3212
- CVE-2023-3268
- CVE-2023-3446
- CVE-2023-3609
- CVE-2023-3611
- CVE-2023-3772
- CVE-2023-3817
- CVE-2023-4016
- CVE-2023-4128
- CVE-2023-4132
- CVE-2023-4155
- CVE-2023-4206
- CVE-2023-4207
- CVE-2023-4208
- CVE-2023-4641
- CVE-2023-4732
- CVE-2023-5678
- CVE-2023-22745

- CVE-2023-23455
- CVE-2023-26545
- CVE-2023-28328
- CVE-2023-28772
- CVE-2023-30456
- CVE-2023-31084
- CVE-2023-31436
- CVE-2023-31486
- CVE-2023-33203
- CVE-2023-33951
- CVE-2023-33952
- CVE-2023-35823
- CVE-2023-35824
- CVE-2023-35825
- CVE-2023-38037
- CVE-2024-0443

1.2.4. Logging 5.7.9

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.7.9 가 포함되어 있습니다.

1.2.4.1. 버그 수정

- 이번 수정 이전에는 자리 표시자에 대해 호스트 또는 여러 호스트를 평가한 후 IPv6 주소가 올바르게 구문 분석되지 않았습니다. 이번 업데이트를 통해 IPv6 주소가 올바르게 구문 분석됩니다. ([LOG-4281](#))
- 이번 업데이트 이전에는 IPv4 전용 노드에서 벡터를 시작하지 못했습니다. 결과적으로 다음 오류와 함께 지표 끝점에 대한 리스너를 생성하지 못했습니다. **Failed to start Prometheus exporter: TCP bind failed: Address family not supported by protocol (os error 97)**. 이번 업데이트를 통해 벡터는 IPv4 전용 노드에서 정상적으로 작동합니다. ([LOG-4589](#))
- 이번 업데이트 이전에는 인덱스 패턴을 생성하는 프로세스 중에 각 로그 출력의 초기 인덱스에서 기본 별칭이 누락되었습니다. 결과적으로 Kibana 사용자는 OpenShift Elasticsearch Operator를 사용하여 인덱스 패턴을 생성할 수 없었습니다. 이번 업데이트에서는 OpenShift Elasticsearch Operator에 누락된 별칭이 추가되어 문제를 해결합니다. Kibana 사용자는 **{app,infra,audit}-000001** 인덱스를 포함하는 인덱스 패턴을 생성할 수 있습니다. ([LOG-4806](#))
- 이번 업데이트 이전에는 Loki Operator에서 사용자 정의 CA 번들을 룰러 Pod에 마운트 하지 않았습니다. 결과적으로 경고 또는 레코딩 규칙을 평가하는 프로세스 중에 오브젝트 스토리지 액세스가 실패했습니다. 이번 업데이트를 통해 Loki Operator는 모든 룰러 Pod에 사용자 정의 CA 번들

을 마운트합니다. 규칙자 Pod는 오브젝트 스토리지에서 로그를 다운로드하여 경고 또는 레코딩 규칙을 평가할 수 있습니다. ([LOG-4837](#))

- 이번 업데이트 이전에는 시간 범위 또는 심각도와 같은 제어를 사용하여 LogQL 쿼리를 변경하면 정규식처럼 레이블 matcher Operator가 변경되었습니다. 이번 업데이트를 통해 쿼리를 업데이트 할 때 정규식 Operator가 변경되지 않은 상태로 유지됩니다. ([LOG-4842](#))
- 이번 업데이트 이전에는 Vector 수집기 배포가 기본 재시도 및 버퍼링 동작에 의존했습니다. 결과적으로 출력 가용성이 불안정한 경우 전달 파이프라인이 모든 메시지를 전달하려고 백업했습니다. 이번 업데이트를 통해 Vector 수집기 배포는 임계값을 초과한 후 메시지 재시도 및 삭제 메시지 수를 제한합니다. ([LOG-4536](#))

1.2.4.2. CVE

- [CVE-2007-4559](#)
- [CVE-2021-43975](#)
- [CVE-2022-3594](#)
- [CVE-2022-3640](#)
- [CVE-2022-4744](#)
- [CVE-2022-28388](#)
- [CVE-2022-38457](#)
- [CVE-2022-40133](#)
- [CVE-2022-40982](#)
- [CVE-2022-41862](#)
- [CVE-2022-42895](#)
- [CVE-2022-45869](#)
- [CVE-2022-45887](#)
- [CVE-2022-48337](#)
- [CVE-2022-48339](#)
- [CVE-2023-0458](#)
- [CVE-2023-0590](#)
- [CVE-2023-0597](#)
- [CVE-2023-1073](#)
- [CVE-2023-1074](#)
- [CVE-2023-1075](#)
- [CVE-2023-1079](#)

- CVE-2023-1118
- CVE-2023-1206
- CVE-2023-1252
- CVE-2023-1382
- CVE-2023-1855
- CVE-2023-1981
- CVE-2023-1989
- CVE-2023-1998
- CVE-2023-2513
- CVE-2023-3138
- CVE-2023-3141
- CVE-2023-3161
- CVE-2023-3212
- CVE-2023-3268
- CVE-2023-3609
- CVE-2023-3611
- CVE-2023-3772
- CVE-2023-4016
- CVE-2023-4128
- CVE-2023-4132
- CVE-2023-4155
- CVE-2023-4206
- CVE-2023-4207
- CVE-2023-4208
- CVE-2023-4641
- CVE-2023-4732
- CVE-2023-22745
- CVE-2023-23455
- CVE-2023-26545

- CVE-2023-28328
- CVE-2023-28772
- CVE-2023-30456
- CVE-2023-31084
- CVE-2023-31436
- CVE-2023-31486
- CVE-2023-32324
- CVE-2023-33203
- CVE-2023-33951
- CVE-2023-33952
- CVE-2023-34241
- CVE-2023-35823
- CVE-2023-35824
- CVE-2023-35825

1.2.5. Logging 5.7.8

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.7.8](#) 이 포함되어 있습니다.

1.2.5.1. 버그 수정

- 이번 업데이트 이전에는 **ClusterLogForwarder** 사용자 정의 리소스(CR)의 **outputRefs** 및 **inputRefs** 매개변수에 동일한 이름을 사용할 때 잠재적인 충돌이 있었습니다. 그 결과 수집기 Pod가 **CrashLoopBackOff** 상태에 진입했습니다. 이번 업데이트를 통해 출력 라벨에 **OUTPUT_** 접두사가 포함되어 출력 라벨과 파이프라인 이름을 구분할 수 있습니다. ([LOG-4383](#))
- 이번 업데이트 이전에는 JSON 로그 구문 분석기를 구성하는 동안 Cluster Logging Operator에 대한 **structuredTypeKey** 또는 **structuredTypeName** 매개변수를 설정하지 않은 경우 잘못된 구성에 대한 경고가 표시되지 않았습니다. 이번 업데이트를 통해 Cluster Logging Operator에 구성 문제를 알려줍니다. ([LOG-4441](#))
- 이번 업데이트 이전에는 Splunk 출력에 지정된 시크릿에 **hecToken** 키가 없거나 잘못된 경우 Vector에서 토큰 없이 Splunk로 로그를 전달했기 때문에 유효성 검사가 실패했습니다. 이번 업데이트를 통해 **hecToken** 키가 없거나 잘못된 경우 **A** 비어 있지 않은 **hecToken** 항목과 함께 검증이 실패합니다. ([LOG-4580](#))
- 이번 업데이트 이전에는 로그의 사용자 정의 시간 범위에서 날짜를 선택하면 웹 콘솔에서 오류가 발생했습니다. 이번 업데이트를 통해 웹 콘솔의 시간 범위 모델에서 날짜를 선택할 수 있습니다. ([LOG-4684](#))

1.2.5.2. CVE

- CVE-2023-40217

- [CVE-2023-44487](#)

1.2.6. 로깅 5.7.7

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.7.7](#) 이 포함되어 있습니다.

1.2.6.1. 버그 수정

- 이번 업데이트 이전에는 FluentD가 EventRouter에서 출력하는 로그를 Vector와 다르게 정규화했습니다. 이번 업데이트를 통해 벡터는 일관된 형식으로 로그 레코드를 생성합니다. ([LOG-4178](#))
- 이번 업데이트 이전에는 최소 버퍼 사용량이 표시되는 것처럼 Cluster Logging Operator가 생성한 지표 대시보드의 **FluentD Buffer Availability** 그래프에 사용된 쿼리에 오류가 발생했습니다. 이번 업데이트를 통해 그래프에는 최대 버퍼 사용량이 표시되고 이제 **FluentD Buffer Usage**로 이름이 변경되었습니다. ([LOG-4555](#))
- 이번 업데이트 이전에는 IPv6 전용 또는 듀얼 스택 OpenShift Container Platform 클러스터에 LokiStack을 배포하면 LokiStack 멤버 목록 등록이 실패했습니다. 그 결과 배포자 Pod가 크래시 루프에 진입했습니다. 이번 업데이트를 통해 관리자는 **lokistack.spec.hashRing.memberlist.enableIpv6**: 값을 **true**로 설정하여 IPv6를 활성화하여 문제를 해결할 수 있습니다. ([LOG-4569](#))
- 이번 업데이트 이전에는 로그 수집기가 컨테이너 로그 행을 읽기 위해 기본 구성 설정에 의존했습니다. 그 결과 로그 수집기에서 순환된 파일을 효율적으로 읽지 않았습니다. 이번 업데이트를 통해 로그 수집기에서 순환된 파일을 효율적으로 처리할 수 있는 바이트 읽기 수가 증가합니다. ([LOG-4575](#))
- 이번 업데이트 이전에는 이벤트 라우터에서 사용되지 않는 메트릭으로 인해 과도한 메모리 사용량으로 인해 컨테이너가 실패했습니다. 이번 업데이트를 통해 사용되지 않은 메트릭을 제거하여 이벤트 라우터의 메모리 사용량이 줄어듭니다. ([LOG-4686](#))

1.2.6.2. CVE

- [CVE-2023-0800](#)
- [CVE-2023-0801](#)
- [CVE-2023-0802](#)
- [CVE-2023-0803](#)
- [CVE-2023-0804](#)
- [CVE-2023-2002](#)
- [CVE-2023-3090](#)
- [CVE-2023-3390](#)
- [CVE-2023-3776](#)
- [CVE-2023-4004](#)
- [CVE-2023-4527](#)
- [CVE-2023-4806](#)

- CVE-2023-4813
- CVE-2023-4863
- CVE-2023-4911
- CVE-2023-5129
- CVE-2023-20593
- CVE-2023-29491
- CVE-2023-30630
- CVE-2023-35001
- CVE-2023-35788

1.2.7. 로깅 5.7.6

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.7.6 이 포함되어 있습니다.

1.2.7.1. 버그 수정

- 이번 업데이트 이전에는 수집기가 컨테이너 로그 행을 읽기 위해 기본 구성 설정에 의존했습니다. 그 결과 수집기에서 교체된 파일을 효율적으로 읽지 않았습니다. 이번 업데이트를 통해 바이트 읽기 수가 증가하여 수집기에서 순환된 파일을 효율적으로 처리할 수 있습니다. ([LOG-4501](#))
- 이번 업데이트 이전에는 사용자가 사전 정의된 필터를 사용하여 URL을 붙여 넣을 때 일부 필터가 반영되지 않았습니다. 이번 업데이트를 통해 UI는 URL의 모든 필터를 반영합니다. ([LOG-4459](#))
- 이번 업데이트 이전에는 사용자 정의 라벨을 사용하여 Loki로 전달하면 Fluentd에서 Vector로 전환할 때 오류가 발생했습니다. 이번 업데이트를 통해 Vector 구성은 Fluentd와 동일한 방식으로 라벨을 종료하여 수집기가 레이블을 시작하고 올바르게 처리하도록 합니다. ([LOG-4460](#))
- 이번 업데이트 이전에는 Observability Logs 콘솔 검색 필드에서 이스케이프해야 하는 특수 문자를 허용하지 않았습니다. 이번 업데이트를 통해 쿼리에서 특수 문자를 올바르게 이스케이프합니다. ([LOG-4456](#))
- 이번 업데이트 이전에는 Splunk로 로그를 보내는 동안 다음 경고 메시지가 표시 되었습니다. **Timestamp**를 찾을 수 없었습니다. 이번 업데이트를 통해 변경 사항은 Timestamp를 검색하는 데 사용되는 로그 필드의 이름을 재정의하고 경고 없이 Splunk로 보냅니다. ([LOG-4413](#))
- 이번 업데이트 이전에는 벡터의 CPU 및 메모리 사용량이 시간이 지남에 따라 증가하고 있었습니다. 이번 업데이트를 통해 이제 지표의 수명을 제한하고 관련 CPU 사용량 및 메모리 풋프린트를 제한하기 위해 Vector 구성에 **expire_metrics_secs=60** 설정이 포함됩니다. ([LOG-4171](#))
- 이번 업데이트 이전에는 LokiStack 게이트웨이가 인증된 요청을 매우 광범위하게 캐시했습니다. 이로 인해 잘못된 권한 부여 결과가 발생했습니다. 이번 업데이트를 통해 LokiStack 게이트웨이 캐시는 이 문제를 보다 세분화하여 해결합니다. ([LOG-4393](#))
- 이번 업데이트 이전에는 Fluentd 런타임 이미지에 런타임 시 필요하지 않은 빌더 툴이 포함되었습니다. 이번 업데이트를 통해 빌더 툴이 제거되어 문제를 해결합니다. ([LOG-4467](#))

1.2.7.2. CVE

- [CVE-2023-3899](#)
- [CVE-2023-4456](#)
- [CVE-2023-32360](#)
- [CVE-2023-34969](#)

1.2.8. 로깅 5.7.4

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.7.4](#) 가 포함되어 있습니다.

1.2.8.1. 버그 수정

- 이번 업데이트 이전에는 CloudMonitor로 로그를 전달할 때 **namespaceUUID** 값이 **logGroupName** 필드에 추가되지 않았습니다. 이번 업데이트를 통해 CloudWatch의 **namespaceUUID** 값이 포함되므로 CloudWatch의 **logGroupName** 이 **logGroupName**으로 표시됩니다. **vectorcw.b443fb9e-bd4c-4b6a-b9d3-c0097f9ed286**. ([LOG-2701](#))
- 이번 업데이트 이전에는 HTTP를 클러스터 외부 대상으로 전달할 때 프록시 URL에 올바른 인증 정보가 제공되어도 벡터 수집기에서 클러스터 전체 HTTP 프록시에 인증할 수 없었습니다. 이번 업데이트를 통해 Vector 로그 수집기에서 클러스터 전체 HTTP 프록시를 인증할 수 있습니다. ([LOG-3381](#))
- 이번 업데이트 이전에는 이 구성이 지원되지 않기 때문에 Fluentd 수집기가 Splunk를 사용하여 출력으로 구성된 경우 Operator가 실패했습니다. 이번 업데이트를 통해 구성 검증에서 지원되지 않는 출력을 거부하여 문제를 해결합니다. ([LOG-4237](#))
- 이번 업데이트 이전에는 AWS Cloudwatch 로그의 TLS 구성에서 Vector 수집기가 활성화된 = **true** 값을 업데이트하고 GCP Stackdriver로 인해 구성 오류가 발생했습니다. 이번 업데이트를 통해 이러한 출력에 대해 **enabled = true** 값이 제거되어 문제를 해결합니다. ([LOG-4242](#))
- 이번 업데이트 이전에는 로그에서 Vector 수집기에서 다음과 같은 오류 메시지를 패닉시킬 수 있습니다. '모든 분기에서 'vector-worker' panicked가 비활성화되고 다른 분기', **src/kubernetes/reflector.rs:26:9** 입니다. 이번 업데이트를 통해 오류가 해결되었습니다. ([LOG-4275](#))
- 이번 업데이트 이전에는 Operator가 해당 태넌트에 대한 추가 옵션으로 구성된 경우 Loki Operator의 문제로 인해 애플리케이션 태넌트에 대한 **alert-manager** 구성이 사라졌습니다. 이번 업데이트를 통해 생성된 Loki 구성에 사용자 정의 및 자동 생성 구성이 모두 포함됩니다. ([LOG-4361](#))
- 이번 업데이트 이전에는 AWS Cloudwatch 전달에서 STS를 사용하여 여러 역할을 인증하는 데 사용된 경우 최근 업데이트로 인해 인증 정보가 고유하지 않았습니다. 이번 업데이트를 통해 STS 역할 및 정적 인증 정보의 여러 조합을 다시 사용하여 AWS Cloudwatch로 인증할 수 있습니다. ([LOG-4368](#))
- 이번 업데이트 이전에는 Loki에서 활성 스트림의 레이블 값을 필터링하지만 중복은 제거되지 않아 Grafana의 Label Browser를 사용할 수 없게 되었습니다. 이번 업데이트를 통해 Loki는 활성 스트림에 대해 중복 레이블 값을 필터링하여 문제를 해결합니다. ([LOG-4389](#))
- **ClusterLogForwarder** 사용자 정의 리소스(CR)에 지정된 **name** 필드가 없는 파이프라인은 OpenShift Logging 5.7으로 업그레이드한 후 작업을 중지했습니다. 이번 업데이트를 통해 오류가 해결되었습니다. ([LOG-4120](#))

1.2.8.2. CVE

- [CVE-2022-25883](#)
- [CVE-2023-22796](#)

1.2.9. 로깅 5.7.3

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.7.3](#) 이 포함되어 있습니다.

1.2.9.1. 버그 수정

- 이번 업데이트 이전에는 OpenShift Container Platform 웹 콘솔 내에서 로그를 볼 때 캐시된 파일로 인해 데이터가 새로 고쳐지지 않았습니다. 이번 업데이트를 통해 부트스트랩 파일이 캐시되지 않아 문제를 해결합니다. ([LOG-4100](#))
- 이번 업데이트 이전에는 Loki Operator가 구성 문제를 확인하기 어려운 방식으로 오류를 재설정했습니다. 이번 업데이트를 통해 구성 오류가 해결될 때까지 오류가 지속됩니다. ([LOG-4156](#))
- 이번 업데이트 이전에는 **RulerConfig** CR(사용자 정의 리소스)을 변경한 후 LokiStack 룰러가 다시 시작되지 않았습니다. 이번 업데이트를 통해 Loki Operator는 **RulerConfig** CR을 업데이트한 후 룰러 Pod를 다시 시작합니다. ([LOG-4161](#))
- 이번 업데이트 이전에는 입력 일치 레이블 값에 **ClusterLogForwarder** 내에 / 문자가 포함된 경우 백터 수집기가 예기치 않게 종료되었습니다. 이번 업데이트에서는 match 레이블을 quoting하여 수집기가 로그를 시작하고 수집 할 수 있도록 하여 문제를 해결합니다. ([LOG-4176](#))
- 이번 업데이트 이전에는 **LokiStack** CR이 테넌트 제한을 정의했지만 글로벌 제한이 아닌 경우 Loki Operator가 예기치 않게 종료되었습니다. 이번 업데이트를 통해 Loki Operator는 글로벌 제한 없이 **LokiStack** CR을 처리하여 문제를 해결할 수 있습니다. ([LOG-4198](#))
- 이번 업데이트 이전에는 제공된 개인 키가 암호로 보호되는 경우 Fluentd에서 Elasticsearch 클러스터에 로그를 보내지 않았습니다. 이번 업데이트를 통해 Elasticsearch와의 연결을 설정 할 때 Fluentd가 암호로 보호된 개인 키를 올바르게 처리합니다. ([LOG-4258](#))
- 이번 업데이트 이전에는 네임스페이스가 8,000개 이상인 클러스터로 인해 네임스페이스 목록이 **http.max_header_size** 설정보다 크므로 Elasticsearch가 쿼리를 거부했습니다. 이번 업데이트를 통해 헤더 크기의 기본값이 증가하여 문제를 해결합니다. ([LOG-4277](#))
- 이번 업데이트 이전에는 **ClusterLogForwarder** CR 내에 / 문자가 포함된 라벨 값으로 인해 수집기가 예기치 않게 종료되었습니다. 이번 업데이트를 통해 슬래시가 밑줄로 교체되어 문제를 해결합니다. ([LOG-4095](#))
- 이번 업데이트 이전에는 관리되지 않는 상태로 설정된 경우 Cluster Logging Operator가 예기치 않게 종료되었습니다. 이번 업데이트를 통해 **ClusterLogForwarder** CR 조정을 시작하기 전에 **ClusterLogging** 리소스가 올바른 관리 상태에 있는지 확인하고 문제를 해결합니다. ([LOG-4177](#))
- 이번 업데이트 이전에는 OpenShift Container Platform 웹 콘솔 내에서 로그를 볼 때 히스토그램을 통해 드래그하여 시간 범위를 선택하면 Pod 세부 정보 내부의 집계된 로그 뷰에서 작동하지 않았습니다. 이번 업데이트를 통해 이 뷰의 히스토그램을 드래그하여 시간 범위를 선택할 수 있습니다. ([LOG-4108](#))
- 이번 업데이트 이전에는 OpenShift Container Platform 웹 콘솔 내에서 로그를 볼 때 30초 이상 시간 초과를 큐리합니다. 이번 업데이트를 통해 configmap/logging-view-plugin에 시간 초과 값 을 구성할 수 있습니다. ([LOG-3498](#))

- 이번 업데이트 이전에는 OpenShift Container Platform 웹 콘솔 내에서 로그를 볼 때 사용 가능한 추가 옵션을 클릭하면 처음 클릭할 때만 더 많은 로그 항목이 로드되었습니다. 이번 업데이트를 통해 각 클릭마다 더 많은 항목이 로드됩니다. ([OU-188](#))
- 이번 업데이트 이전에는 OpenShift Container Platform 웹 콘솔 내에서 로그를 볼 때 **streaming** 옵션을 클릭하면 실제 로그를 표시하지 않고 **streaming** 로그만 표시됩니다. 이번 업데이트를 통해 메시지와 로그 스트림이 모두 올바르게 표시됩니다. ([OU-166](#))

1.2.9.2. CVE

- [CVE-2020-24736](#)
- [CVE-2022-48281](#)
- [CVE-2023-1667](#)
- [CVE-2023-2283](#)
- [CVE-2023-24329](#)
- [CVE-2023-26115](#)
- [CVE-2023-26136](#)
- [CVE-2023-26604](#)
- [CVE-2023-28466](#)

1.2.10. 로깅 5.7.2

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.7.2](#) 가 포함되어 있습니다.

1.2.10.1. 버그 수정

- 이번 업데이트 이전에는 보류 중인 종료자가 있기 때문에 **openshift-logging** 네임스페이스를 직접 삭제할 수 없었습니다. 이번 업데이트를 통해 종료자가 더 이상 사용되지 않으므로 네임스페이스를 직접 삭제할 수 있습니다. ([LOG-3316](#))
- 이번 업데이트 이전에는 OpenShift Container Platform 설명서에 따라 변경된 경우 **run.sh** 스크립트에서 잘못된 **chunk_limit_size** 값을 표시했습니다. 그러나 환경 변수 **\$BUFFER_SIZE_LIMIT**를 통해 **chunk_limit_size**를 설정할 때 스크립트에 올바른 값이 표시되었습니다. 이번 업데이트를 통해 이제 **run.sh** 스크립트에서 두 시나리오에 올바른 **chunk_limit_size** 값을 일관되게 표시합니다. ([LOG-3330](#))
- 이번 업데이트 이전에는 OpenShift Container Platform 웹 콘솔의 로깅 보기 플러그인에서 사용자 정의 노드 배치 또는 허용 오차를 허용하지 않았습니다. 이번 업데이트에서는 로깅 보기 플러그인에 대한 노드 배치 및 허용 오차를 정의하는 기능이 추가되었습니다. ([LOG-3749](#))
- 이번 업데이트 이전에는 Fluentd HTTP 플러그인을 통해 DataDog에 로그를 보낼 때 Cluster Logging Operator에 Unsupported Media Type 예외가 발생했습니다. 이번 업데이트를 통해 사용자는 HTTP 헤더 Content-Type을 구성하여 로그 전달을 위해 콘텐츠 유형을 원활하게 할당할 수 있습니다. 제공된 값은 플러그인 내의 **content_type** 매개변수에 자동으로 할당되어 로그 전송에 성공합니다. ([LOG-3784](#))
- 이번 업데이트 이전에는 **ClusterLogForwarder** CR(사용자 정의 리소스)에서 **detectMultilineErrors** 필드가 **true**로 설정된 경우 PHP 다중 줄 오류가 별도의 로그 항목으로 기

록되어 스택 추적이 여러 메시지에 분할되었습니다. 이번 업데이트를 통해 PHP에 대한 다중 줄 오류 탐지가 활성화되어 전체 스택 추적이 단일 로그 메시지에 포함됩니다. ([LOG-3878](#))

- 이번 업데이트 이전에는 이름에 공간이 포함된 **ClusterLogForwarder** 파이프라인으로 인해 Vector 수집기 Pod가 지속적으로 충돌했습니다. 이번 업데이트를 통해 파이프라인 이름의 모든 공백, 대시(-) 및 점(.)이 밑줄(_)으로 교체됩니다. ([LOG-3945](#))
- 이번 업데이트 이전에는 **log_forwarder_output** 지표에 **http** 매개변수가 포함되지 않았습니다. 이번 업데이트에서는 누락된 매개변수가 메트릭에 추가되었습니다. ([LOG-3997](#))
- 이번 업데이트 이전에는 콜론으로 종료될 때 Fluentd에서 일부 다중 줄 JavaScript 클라이언트 예외를 식별하지 못했습니다. 이번 업데이트를 통해 Fluentd 버퍼 이름 앞에 밑줄이 표시되어 문제를 해결합니다. ([LOG-4019](#))
- 이번 업데이트 이전에는 페이로드의 키와 일치하는 Kafka 출력 항목에 기록하도록 로그 전달을 구성할 때 오류로 인해 로그가 삭제되었습니다. 이번 업데이트를 통해 Fluentd의 버퍼 이름 앞에 밑줄이 지정되어 문제를 해결합니다. ([LOG-4027](#))
- 이번 업데이트 이전에는 LokiStack 게이트웨이에서 사용자의 액세스 권한을 적용하지 않고 네임스페이스의 레이블 값을 반환했습니다. 이번 업데이트를 통해 LokiStack 게이트웨이는 값 요청에 레이블을 지정하여 문제를 해결합니다. ([LOG-4049](#))
- 이번 업데이트 이전에는 **tls.insecureSkipVerify** 옵션이 **true**로 설정된 경우 Cluster Logging Operator API에 시크릿에서 제공할 인증서가 필요했습니다. 이번 업데이트를 통해 Cluster Logging Operator API에서 이러한 경우 시크릿에서 더 이상 인증서를 제공할 필요가 없습니다. Operator의 CR에 다음 구성이 추가되었습니다.

```
tls.verify_certificate = false
tls.verify_hostname = false
```

([LOG-3445](#))

- 이번 업데이트 이전에는 LokiStack 경로 구성으로 인해 30초 이상 실행되는 쿼리가 시간 초과되었습니다. 이번 업데이트를 통해 LokiStack 글로벌 및 테넌트별 쿼리Timeout 설정은 경로 시간 초과 설정에 영향을 미치므로 문제를 해결합니다. ([LOG-4052](#))
- 이번 업데이트 이전에는 **collection.type** 기본값을 제거하기 전에 수정되어 Operator에서 더 이상 리소스, 노드 선택, 허용 오차에 대해 더 이상 사용되지 않는 사양을 준수하지 않았습니다. 이번 업데이트에서는 컬렉션보다 **collection.logs** 사양을 항상 선호하도록 Operator 동작을 수정합니다. 이는 기본 필드와 더 이상 사용되지 않는 필드를 모두 사용하여 허용되는 이전 동작과 다르지만, **collection.type** 이 채워지면 더 이상 사용되지 않는 필드는 무시됩니다. ([LOG-4185](#))
- 이번 업데이트 이전에는 출력에 브로커 URL이 지정되지 않은 경우 벡터 로그 수집기에서 여러 Kafka 브로커로 로그를 전달하기 위한 TLS 구성을 생성하지 않았습니다. 이번 업데이트를 통해 여러 브로커에 대해 TLS 구성이 적절하게 생성됩니다. ([LOG-4163](#))
- 이번 업데이트 이전에는 Kafka로의 로그 전달을 위해 암호를 활성화하는 옵션을 사용할 수 없었습니다. 이 제한은 민감한 정보를 잠재적으로 노출할 수 있으므로 보안 위협이 발생했습니다. 이번 업데이트를 통해 이제 Kafka로 로그 전달을 위해 암호를 활성화할 수 있는 원활한 옵션이 제공됩니다. ([LOG-3314](#))
- 이번 업데이트 이전에는 Vector 로그 수집기에서 발신 TLS 연결에 대한 **tlsSecurityProfile** 설정을 준수하지 않았습니다. 이번 업데이트 후 벡터는 TLS 연결 설정을 적절하게 처리합니다. ([LOG-4011](#))

- 이번 업데이트 이전에는 **log_forwarder_output_info** 메트릭에 사용 가능한 모든 출력 유형이 포함되지 않았습니다. 이번 업데이트를 통해 메트릭에는 이전에 누락된 Splunk 및 Google Cloud Logging 데이터가 포함되어 있습니다. ([LOG-4098](#))
- 이번 업데이트 이전에는 **follow_inodes** 가 **true** 로 설정된 경우 Fluentd 수집기가 파일 교체 시 충돌할 수 있었습니다. 이번 업데이트를 통해 **follow_inodes** 설정이 수집기가 충돌하지 않습니다. ([LOG-4151](#))
- 이번 업데이트 이전에는 Fluentd 수집기에서 해당 파일이 추적되는 방식으로 모니터링해야 하는 파일을 잘못 닫을 수 있었습니다. 이번 업데이트를 통해 추적 매개변수가 수정되었습니다. ([LOG-4149](#))
- 이번 업데이트 이전에는 Vector 수집기로 로그를 전달하고 **ClusterLogForwarder** 인스턴스 감사, 애플리케이션 또는 인프라에서 파이프라인 이름을 지정하여 수집기 로그에서 다음 오류와 함께 수집기 Pod가 **CrashLoopBackOff** 상태를 유지했습니다.

ERROR vector::cli: Configuration error. error=redefinition of table transforms.audit for key transforms.audit

이번 업데이트 후 파이프라인 이름은 예약된 입력 이름과 충돌하지 않으며 파이프라인의 이름을 **audit, application** 또는 **infrastructure**로 지정할 수 있습니다. ([LOG-4218](#))

- 이번 업데이트 이전에는 Vector 수집기를 사용하여 로그를 syslog 대상으로 전달하고 **addLogSource** 플래그를 **true**로 설정할 때 **namespace_name=**, **container_name=**, **pod_name=**이라는 전달된 메시지에 다음과 같은 빈 필드가 추가되었습니다. 이번 업데이트를 통해 이러한 필드는 더 이상 저널 로그에 추가되지 않습니다. ([LOG-4219](#))
- 이번 업데이트 이전에는 **structuredTypeKey**를 찾을 수 없고 **structuredTypeName**이 지정되지 않은 경우 로그 메시지가 구조화된 오브젝트로 계속 구문 분석되었습니다. 이번 업데이트를 통해 로그 구문 분석이 예상대로 수행됩니다. ([LOG-4220](#))

1.2.10.2. CVE

- [CVE-2021-26341](#)
- [CVE-2021-33655](#)
- [CVE-2021-33656](#)
- [CVE-2022-1462](#)
- [CVE-2022-1679](#)
- [CVE-2022-1789](#)
- [CVE-2022-2196](#)
- [CVE-2022-2663](#)
- [CVE-2022-3028](#)
- [CVE-2022-3239](#)
- [CVE-2022-3522](#)
- [CVE-2022-3524](#)

- CVE-2022-3564
- CVE-2022-3566
- CVE-2022-3567
- CVE-2022-3619
- CVE-2022-3623
- CVE-2022-3625
- CVE-2022-3627
- CVE-2022-3628
- CVE-2022-3707
- CVE-2022-3970
- CVE-2022-4129
- CVE-2022-20141
- CVE-2022-25147
- CVE-2022-25265
- CVE-2022-30594
- CVE-2022-36227
- CVE-2022-39188
- CVE-2022-39189
- CVE-2022-41218
- CVE-2022-41674
- CVE-2022-42703
- CVE-2022-42720
- CVE-2022-42721
- CVE-2022-42722
- CVE-2022-43750
- CVE-2022-47929
- CVE-2023-0394
- CVE-2023-0461
- CVE-2023-1195

- [CVE-2023-1582](#)
- [CVE-2023-2491](#)
- [CVE-2023-22490](#)
- [CVE-2023-23454](#)
- [CVE-2023-23946](#)
- [CVE-2023-25652](#)
- [CVE-2023-25815](#)
- [CVE-2023-27535](#)
- [CVE-2023-29007](#)

1.2.11. 로깅 5.7.1

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.7.1](#) 가 포함되어 있습니다.

1.2.11.1. 버그 수정

- 이번 업데이트 이전에는 Cluster Logging Operator Pod 로그 내에 다양한 주의 메시지가 있으면 로그 가독성이 감소하고 중요한 시스템 이벤트를 식별하는 데 어려움이 발생했습니다. 이번 업데이트를 통해 Cluster Logging Operator Pod 로그 내에서 noisy 메시지를 크게 줄임으로써 문제가 해결됩니다. ([LOG-3482](#))
- 이번 업데이트 이전에는 사용자 정의 리소스가 다른 값을 사용한 경우에도 API 서버에서 **CollectorSpec.Type** 필드의 값을 벡터로 재설정합니다. 이번 업데이트에서는 **CollectorSpec.Type** 필드의 기본값을 제거하여 이전 동작을 복원합니다. ([LOG-4086](#))
- 이번 업데이트 이전에는 로그 히스토그램을 클릭하고 끌어서 OpenShift Container Platform 웹 콘솔에서 시간 범위를 선택할 수 없었습니다. 이번 업데이트를 통해 클릭 및 드래그를 사용하여 시간 범위를 성공적으로 선택할 수 있습니다. ([LOG-4501](#))
- 이번 업데이트 이전에는 OpenShift Container Platform 웹 콘솔의 리소스 표시 링크를 클릭하면 적용되지 않았습니다. 이번 업데이트를 통해 "Forwarded Resources" 링크의 기능을 수정하여 각 로그 항목에 대한 리소스 표시를 전환하여 문제가 해결됩니다. ([LOG-3218](#))

1.2.11.2. CVE

- [CVE-2023-21930](#)
- [CVE-2023-21937](#)
- [CVE-2023-21938](#)
- [CVE-2023-21939](#)
- [CVE-2023-21954](#)
- [CVE-2023-21967](#)
- [CVE-2023-21968](#)

- CVE-2023-28617

1.2.12. Logging 5.7.0

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.7.0](#) 이 포함되어 있습니다.

1.2.12.1. 기능 개선

이번 업데이트를 통해 로깅을 활성화하여 여러 줄 예외를 감지하고 단일 로그 항목으로 재조정할 수 있습니다.

로깅을 사용하여 다중 줄 예외를 감지하고 단일 로그 항목으로 재조정할 수 있도록 하려면 **ClusterLogForwarder** 사용자 정의 리소스(CR)에 값이 **true**인 **detectMultilineErrors** 필드가 포함되어 있는지 확인합니다.

1.2.12.2. 확인된 문제

없음.

1.2.12.3. 버그 수정

- 이번 업데이트 이전에는 CloudEventStack의 게이트웨이 구성 요소에 대한 **nodeSelector** 속성이 노드 예약에 영향을 미치지 않았습니다. 이번 업데이트를 통해 **nodeSelector** 특성이 예상대로 작동합니다. ([LOG-3713](#))

1.2.12.4. CVE

- CVE-2023-1999
- CVE-2023-28617

1.3. 로깅 5.6



참고

로깅은 핵심 OpenShift Container Platform과 별도의 릴리스 주기와 함께 설치 가능한 구성 요소로 제공됩니다. [Red Hat OpenShift Container Platform 라이프 사이클 정책](#)은 릴리스 호환성에 대해 간단히 설명합니다.



참고

stable 채널은 최신 로깅 릴리스에 대한 업데이트만 제공합니다. 이전 릴리스에 대한 업데이트를 계속 받으려면 서브스크립션 채널을 **stable-x.y**로 변경해야 합니다. 여기서 **x.y**는 설치한 로깅 및 마이너 버전을 나타냅니다. 예를 들면 **stable-5.7**입니다.

1.3.1. Logging 5.6.17

이 릴리스에는 [OpenShift Logging 버그 수정 5.6.17](#)이 포함되어 있습니다.

1.3.1.1. 버그 수정

- 이번 업데이트 이전에는 Red Hat 빌드 파이프라인에서 Loki 빌드의 기존 빌드 세부 정보와 revision, 분기, 버전과 같은 생략된 정보를 사용하지 않았습니다. 이번 업데이트를 통해 Red Hat 빌드 파이프라인에서 Loki 빌드에 이러한 세부 정보를 추가하여 문제를 해결합니다. ([LOG-5203](#))
- 이번 업데이트 이전에는 Loki Operator의 **ServiceMonitor** 구성이 여러 Kubernetes 서비스와 일치하여 Loki Operator의 메트릭이 여러 번 수집될 수 있었습니다. 이번 업데이트를 통해 **ServiceMonitor** 설정은 이제 전용 메트릭 서비스와만 일치합니다. ([LOG-5252](#))
- 이번 업데이트 이전에는 빌드 파이프라인에 빌드 날짜에 대한 링커 플래그가 포함되어 있지 않아 Loki 빌드에 **buildDate** 및 **goVersion** 의 빈 문자열이 표시되었습니다. 이번 업데이트를 통해 빌드 파이프라인에 누락된 링커 플래그를 추가하면 문제가 해결되었습니다. ([LOG-5276](#))
- 이번 업데이트 이전에는 **openshift-operators-redhat** 네임스페이스의 Loki Operator **ServiceMonitor**에서 인증에 정적 토큰과 CA 파일을 사용하여 **ServiceMonitor** 구성의 User Workload Monitoring 사양의 Prometheus Operator에 오류가 발생했습니다. 이번 업데이트를 통해 **openshift-operators-redhat** 네임스페이스의 Loki Operator **ServiceMonitor**는 이제 **LocalReference** 오브젝트의 서비스 계정 토큰 시크릿을 참조합니다. 이 방법을 사용하면 Prometheus Operator의 User Workload Monitoring 사양에서 Loki Operator **ServiceMonitor**를 성공적으로 처리할 수 있으므로 Prometheus가 Loki Operator 메트릭을 스크랩할 수 있습니다. ([LOG-5242](#))

1.3.1.2. CVE

- [CVE-2021-35937](#)
- [CVE-2021-35938](#)
- [CVE-2021-35939](#)
- [CVE-2024-24786](#)

1.3.2. 로깅 5.6.16

이 릴리스에는 로깅 버그 수정 5.6.16이 포함되어 있습니다.

1.3.2.1. 버그 수정

- 이번 업데이트 이전에는 사용자 정의 S3 인증 기관을 읽도록 구성된 경우 Loki Operator는 ConfigMap 이름 또는 콘텐츠가 변경된 경우 구성을 자동으로 업데이트하지 않습니다. 이번 업데이트를 통해 Loki Operator에서 ConfigMap에 대한 변경 사항을 조사하고 생성된 구성을 자동으로 업데이트합니다. ([LOG-4967](#))

1.3.2.2. CVE

1.3.3. 로깅 5.6.15

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.6.15 가 포함되어 있습니다.

1.3.3.1. 버그 수정

이번 업데이트 이전에는 LokiStack 규칙의 Pod에서 교차 Pod 통신에 사용되는 HTTP URL의 IPv6 Pod IP를 포맷하지 않아 Prometheus 호환 API를 통한 규칙 및 경고를 쿼리할 수 없었습니다. 이번 업데이트를 통해 LokiStack 룰러 Pod는 IPv6 포드 IP를 대괄호로 캡슐화하여 문제를 해결합니다. ([LOG-4892](#))

1.3.3.2. CVE

- [CVE-2021-3468](#)
- [CVE-2023-3446](#)
- [CVE-2023-3817](#)
- [CVE-2023-5678](#)
- [CVE-2023-38469](#)
- [CVE-2023-38470](#)
- [CVE-2023-38471](#)
- [CVE-2023-38472](#)
- [CVE-2023-38473](#)

1.3.4. Logging 5.6.14

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.6.14](#) 가 포함되어 있습니다.

1.3.4.1. 버그 수정

- 이번 업데이트 이전에는 인덱스 패턴을 생성하는 프로세스 중에 각 로그 출력의 초기 인덱스에서 기본 별칭이 누락되었습니다. 결과적으로 Kibana 사용자는 OpenShift Elasticsearch Operator를 사용하여 인덱스 패턴을 생성할 수 없었습니다. 이번 업데이트에서는 OpenShift Elasticsearch Operator에 누락된 별칭이 추가되어 문제를 해결합니다. Kibana 사용자는 **{app,infra,audit}-000001** 인덱스를 포함하는 인덱스 패턴을 생성할 수 있습니다. ([LOG-4807](#))
- 이번 업데이트 이전에는 Loki Operator에서 사용자 정의 CA 범들을 룰러 Pod에 마운트하지 않았습니다. 결과적으로 경고 또는 레코딩 규칙을 평가하는 프로세스 중에 오브젝트 스토리지 액세스가 실패했습니다. 이번 업데이트를 통해 Loki Operator는 모든 룰러 Pod에 사용자 정의 CA 범들을 마운트합니다. 규칙자 Pod는 오브젝트 스토리지에서 로그를 다운로드하여 경고 또는 레코딩 규칙을 평가할 수 있습니다. ([LOG-4838](#))

1.3.4.2. CVE

- [CVE-2007-4559](#)
- [CVE-2021-43975](#)
- [CVE-2022-3594](#)
- [CVE-2022-3640](#)
- [CVE-2022-4744](#)
- [CVE-2022-28388](#)
- [CVE-2022-38457](#)
- [CVE-2022-40133](#)

- CVE-2022-40982
- CVE-2022-41862
- CVE-2022-42895
- CVE-2022-45869
- CVE-2022-45887
- CVE-2022-48337
- CVE-2022-48339
- CVE-2023-0458
- CVE-2023-0590
- CVE-2023-0597
- CVE-2023-1073
- CVE-2023-1074
- CVE-2023-1075
- CVE-2023-1079
- CVE-2023-1118
- CVE-2023-1206
- CVE-2023-1252
- CVE-2023-1382
- CVE-2023-1855
- CVE-2023-1981
- CVE-2023-1989
- CVE-2023-1998
- CVE-2023-2513
- CVE-2023-3138
- CVE-2023-3141
- CVE-2023-3161
- CVE-2023-3212
- CVE-2023-3268
- CVE-2023-3609

- CVE-2023-3611
- CVE-2023-3772
- CVE-2023-4016
- CVE-2023-4128
- CVE-2023-4132
- CVE-2023-4155
- CVE-2023-4206
- CVE-2023-4207
- CVE-2023-4208
- CVE-2023-4641
- CVE-2023-4732
- CVE-2023-22745
- CVE-2023-23455
- CVE-2023-26545
- CVE-2023-28328
- CVE-2023-28772
- CVE-2023-30456
- CVE-2023-31084
- CVE-2023-31436
- CVE-2023-31486
- CVE-2023-32324
- CVE-2023-33203
- CVE-2023-33951
- CVE-2023-33952
- CVE-2023-34241
- CVE-2023-35823
- CVE-2023-35824
- CVE-2023-35825

1.3.5. Logging 5.6.13

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.6.13 이 포함되어 있습니다.

1.3.5.1. 버그 수정

없음.

1.3.5.2. CVE

- [CVE-2023-40217](#)
- [CVE-2023-44487](#)

1.3.6. 로깅 5.6.12

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.6.12 가 포함되어 있습니다.

1.3.6.1. 버그 수정

- 이번 업데이트 이전에는 IPv6 전용 또는 듀얼 스택 OpenShift Container Platform 클러스터에 LokiStack을 배포하면 LokiStack 멤버 목록 등록이 실패했습니다. 그 결과 배포자 Pod가 크래시 루프에 진입했습니다. 이번 업데이트를 통해 관리자는 **lokistack.spec.hashRing.memberlist.enableIpv6**: 값을 **true**로 설정하여 IPv6를 활성화하여 문제를 해결할 수 있습니다. 현재 IPv6 지원 클러스터에서는 로그 경고를 사용할 수 없습니다. ([LOG-4570](#))
- 이번 업데이트 이전에는 최소 버퍼 사용량이 표시되는 것처럼 Cluster Logging Operator가 생성한 지표 대시보드의 **FluentD Buffer Availability** 그래프에 사용된 쿼리에 오류가 발생했습니다. 이번 업데이트를 통해 그래프에는 최대 버퍼 사용량이 표시되고 이제 **FluentD Buffer Usage**로 이름이 변경되었습니다. ([LOG-4579](#))
- 이번 업데이트 이전에는 이벤트 라우터에서 사용되지 않는 메트릭으로 인해 과도한 메모리 사용량으로 인해 컨테이너가 실패했습니다. 이번 업데이트를 통해 사용되지 않은 메트릭을 제거하여 이벤트 라우터의 메모리 사용량이 줄어듭니다. ([LOG-4687](#))

1.3.6.2. CVE

- [CVE-2023-0800](#)
- [CVE-2023-0801](#)
- [CVE-2023-0802](#)
- [CVE-2023-0803](#)
- [CVE-2023-0804](#)
- [CVE-2023-2002](#)
- [CVE-2023-3090](#)
- [CVE-2023-3390](#)
- [CVE-2023-3776](#)
- [CVE-2023-4004](#)

- CVE-2023-4527
- CVE-2023-4806
- CVE-2023-4813
- CVE-2023-4863
- CVE-2023-4911
- CVE-2023-5129
- CVE-2023-20593
- CVE-2023-29491
- CVE-2023-30630
- CVE-2023-35001
- CVE-2023-35788

1.3.7. 로깅 5.6.11

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.6.11](#) 이 포함되어 있습니다.

1.3.7.1. 버그 수정

- 이번 업데이트 이전에는 LokiStack 게이트웨이가 인증된 요청을 매우 광범위하게 캐시했습니다. 이로 인해 잘못된 권한 부여 결과가 발생했습니다. 이번 업데이트를 통해 LokiStack 게이트웨이 캐시는 이 문제를 보다 세분화하여 해결합니다. ([LOG-4435](#))

1.3.7.2. CVE

- CVE-2023-3899
- CVE-2023-32360
- CVE-2023-34969

1.3.8. 로깅 5.6.9

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.6.9](#) 가 포함되어 있습니다.

1.3.8.1. 버그 수정

- 이번 업데이트 이전에는 AWS Cloudwatch 전달에서 STS를 사용하여 여러 역할을 인증하는 데 사용된 경우 최근 업데이트로 인해 인증 정보가 고유하지 않았습니다. 이번 업데이트를 통해 STS 역할 및 정적 인증 정보의 여러 조합을 다시 사용하여 AWS Cloudwatch로 인증할 수 있습니다. ([LOG-4084](#))
- 이번 업데이트 이전에는 로그에서 Vector 수집기에서 다음과 같은 오류 메시지를 패닉시킬 수 있습니다. '모든 분기에서 'vector-worker' panicked가 비활성화되고 다른 분기', `src/kubernetes/reflector.rs:26:9` 입니다. 이번 업데이트를 통해 오류가 해결되었습니다. ([LOG-4276](#))

- 이번 업데이트 이전에는 Loki에서 활성 스트림의 레이블 값을 필터링하지만 중복은 제거되지 않아 Grafana의 Label Browser를 사용할 수 없게 되었습니다. 이번 업데이트를 통해 Loki는 활성 스트림에 대해 중복 레이블 값을 필터링하여 문제를 해결합니다. ([LOG-4390](#))

1.3.8.2. CVE

- [CVE-2020-24736](#)
- [CVE-2022-48281](#)
- [CVE-2023-1667](#)
- [CVE-2023-2283](#)
- [CVE-2023-24329](#)
- [CVE-2023-26604](#)
- [CVE-2023-28466](#)
- [CVE-2023-32233](#)

1.3.9. 로깅 5.6.8

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.6.8](#) 이 포함되어 있습니다.

1.3.9.1. 버그 수정

- 이번 업데이트 이전에는 입력 일치 레이블 값에 **ClusterLogForwarder** 내에 / 문자가 포함된 경우 백터 수집기가 예기치 않게 종료되었습니다. 이번 업데이트에서는 match 레이블을 quoting하여 수집기가 로그를 시작하고 수집 할 수 있도록 하여 문제를 해결합니다. ([LOG-4091](#))
- 이번 업데이트 이전에는 OpenShift Container Platform 웹 콘솔 내에서 로그를 볼 때 **사용 가능한** 추가 옵션을 클릭하면 처음 클릭 할 때만 더 많은 로그 항목이 로드되었습니다. 이번 업데이트를 통해 각 클릭마다 더 많은 항목이 로드됩니다. ([OU-187](#))
- 이번 업데이트 이전에는 OpenShift Container Platform 웹 콘솔 내에서 로그를 볼 때 **streaming** 옵션을 클릭하면 실제 로그를 표시하지 않고 **streaming** 로그만 표시됩니다. 이번 업데이트를 통해 메시지와 로그 스트림이 모두 올바르게 표시됩니다. ([OU-189](#))
- 이번 업데이트 이전에는 Loki Operator가 구성 문제를 확인하기 어려운 방식으로 오류를 재설정했습니다. 이번 업데이트를 통해 구성 오류가 해결될 때까지 오류가 지속됩니다. ([LOG-4158](#))
- 이번 업데이트 이전에는 네임스페이스가 8,000개 이상인 클러스터로 인해 네임스페이스 목록이 **http.max_header_size** 설정보다 크므로 Elasticsearch가 쿼리를 거부했습니다. 이번 업데이트를 통해 헤더 크기의 기본값이 증가하여 문제를 해결합니다. ([LOG-4278](#))

1.3.9.2. CVE

- [CVE-2020-24736](#)
- [CVE-2022-48281](#)
- [CVE-2023-1667](#)
- [CVE-2023-2283](#)

- CVE-2023-24329
- CVE-2023-26604
- CVE-2023-28466

1.3.10. Logging 5.6.5

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.6.5](#) 가 포함되어 있습니다.

1.3.10.1. 버그 수정

- 이번 업데이트 이전에는 템플릿 정의로 인해 Elasticsearch에서 일부 레이블과 namespace_labels를 인덱싱할 수 없어 데이터 수집 문제가 발생했습니다. 이번 업데이트를 통해 수정으로 레이블의 점과 슬래시가 교체되어 문제를 효과적으로 해결하기 위해 레이블의 점과 슬래시가 교체됩니다. ([LOG-3419](#))
- 이번 업데이트 이전에는 OpenShift 웹 콘솔의 로그 페이지가 10.0.0.1Stack에 연결하지 못하면 일반적인 오류 메시지가 표시되어 추가 컨텍스트 또는 문제 해결 제안이 제공되지 않았습니다. 이번 업데이트를 통해 보다 구체적인 세부 정보와 문제 해결을 위한 권장 사항을 포함하도록 오류 메시지가 개선되었습니다. ([LOG-3750](#))
- 이번 업데이트 이전에는 시간 범위 형식이 검증되지 않아 사용자 지정 날짜 범위를 선택하는 데 오류가 발생했습니다. 이번 업데이트를 통해 이제 시간 형식이 검증되어 사용자가 유효한 범위를 선택할 수 있습니다. 잘못된 시간 범위 형식을 선택하면 사용자에게 오류 메시지가 표시됩니다. ([LOG-3583](#))
- 이번 업데이트 이전에는 표현식의 길이가 5120자를 초과하지 않더라도, 대부분의 경우 로그를 검색할 때 쿼리가 실패했습니다. 이번 업데이트를 통해 쿼리 권한 부여 라벨 일치자가 최적화되어 문제를 해결합니다. ([LOG-3480](#))
- 이번 업데이트 이전에는 CloudEvent Operator에서 개인 IP에 멤버 목록을 사용할 때 모든 구성 요소를 배치하기에 충분한 멤버 목록 구성을 생성하지 못했습니다. 이번 업데이트를 통해 생성된 구성에 공개된 포트가 포함되어 모든 구성 요소를 성공적으로 조회할 수 있습니다. ([LOG-4008](#))

1.3.10.2. CVE

- CVE-2022-4269
- CVE-2022-4378
- CVE-2023-0266
- CVE-2023-0361
- CVE-2023-0386
- CVE-2023-27539
- CVE-2023-28120

1.3.11. Logging 5.6.4

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.6.4](#) 가 포함되어 있습니다.

1.3.11.1. 버그 수정

- 이번 업데이트 이전에는 로그 저장소로 ScanSettingStack이 배포된 경우 CloudEvent Pod에서 생성한 로그가 수집되어 stack으로 전송되었습니다. 이번 업데이트를 통해 CloudEvent에서 생성된 로그는 컬렉션에서 제외되며 저장되지 않습니다. ([LOG-3280](#))
- 이번 업데이트 이전에는 OpenShift 웹 콘솔의 로그 페이지에 있는 쿼리 편집기가 비어 있으면 드롭다운 메뉴가 채워지지 않았습니다. 이번 업데이트를 통해 빈 쿼리를 시도하면 오류 메시지가 표시되고 드롭다운 메뉴가 예상대로 채워집니다. ([LOG-3454](#))
- 이번 업데이트 이전에는 **tls.insecureSkipVerify** 옵션이 **true**로 설정되면 Cluster Logging Operator가 잘못된 구성을 생성했습니다. 결과적으로 Operator는 인증서 검증을 건너뛰려고 할 때 Elasticsearch에 데이터를 보내지 못했습니다. 이번 업데이트를 통해 **tls.insecureSkipVerify** 이 활성화된 경우에도 Cluster Logging Operator에서 올바른 TLS 구성을 생성합니다. 결과적으로 인증서 검증을 건너뛰는 경우에도 데이터를 Elasticsearch로 성공적으로 전송할 수 있습니다. ([LOG-3475](#))
- 이번 업데이트 이전에는 구조화된 구문 분석이 활성화되고 메시지가 여러 대상으로 전달되면 깊은 복사되지 않았습니다. 이로 인해 구조화된 메시지를 포함하여 수신된 로그 중 일부가 생성되었지만 다른 로그는 그렇지 않았습니다. 이번 업데이트를 통해 JSON 구문 분석 전에 구성 생성이 깊은 복사 메시지로 수정되었습니다. 결과적으로 여러 대상으로 전달되는 경우에도 수신된 모든 메시지에 구조화된 메시지가 포함됩니다. ([LOG-3640](#))
- 이번 업데이트 이전에는 컬렉션 필드에 `{}`이 포함된 경우 Operator가 충돌할 수 있습니다. 이번 업데이트를 통해 Operator는 이 값을 무시하여 Operator가 중단 없이 원활하게 실행할 수 있습니다. ([LOG-3733](#))
- 이번 업데이트 이전에는 CloudEventStack의 게이트웨이 구성 요소에 대한 **nodeSelector** 속성이 적용되지 않았습니다. 이번 업데이트를 통해 **nodeSelector** 특성이 예상대로 작동합니다. ([LOG-3783](#))
- 이번 업데이트 이전에는 정적 ScanSettingStack 멤버 목록 구성은 개인 IP 네트워크에만 의존했습니다. 결과적으로 OpenShift Container Platform 클러스터 Pod 네트워크가 공용 IP 범위로 구성되면 CloudEventStack Pod가 크래시 루프가 발생했습니다. 이번 업데이트를 통해 이제 ScanSettingStack 관리자에게 memberlist 구성에 Pod 네트워크를 사용하는 옵션이 있습니다. 이렇게 하면 OpenShift Container Platform 클러스터 pod 네트워크가 공용 IP 범위로 구성되면 이 문제를 해결하고 stack Pod가 크래시 루프 상태가 되지 않습니다. ([LOG-3814](#))
- 이번 업데이트 이전에는 **tls.insecureSkipVerify** 필드가 **true**로 설정된 경우 Cluster Logging Operator가 잘못된 구성을 생성했습니다. 결과적으로 Operator는 인증서 검증을 생략할 때 Elasticsearch에 데이터를 보내지 못했습니다. 이번 업데이트를 통해 **tls.insecureSkipVerify** 이 활성화된 경우에도 Operator에서 올바른 TLS 구성을 생성합니다. 결과적으로 인증서 검증을 건너뛰는 경우에도 데이터를 Elasticsearch로 성공적으로 전송할 수 있습니다. ([LOG-3838](#))
- 이번 업데이트 이전에는 Elasticsearch Operator 없이 CLO(Cluster Logging Operator)를 설치한 경우 CLO Pod에 Elasticsearch 삭제와 관련된 오류 메시지가 계속 표시되었습니다. 이번 업데이트를 통해 이제 CLO에서 오류 메시지를 표시하기 전에 추가 검사를 수행합니다. 결과적으로 Elasticsearch 삭제와 관련된 오류 메시지가 Elasticsearch Operator가 없으면 더 이상 표시되지 않습니다. ([LOG-3763](#))

1.3.11.2. CVE

- [CVE-2022-4304](#)
- [CVE-2022-4450](#)

- [CVE-2023-0215](#)
- [CVE-2023-0286](#)
- [CVE-2023-0767](#)
- [CVE-2023-23916](#)

1.3.12. Logging 5.6.3

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.6.3](#) 이 포함되어 있습니다.

1.3.12.1. 버그 수정

- 이번 업데이트 이전에는 Operator에서 게이트웨이 테넌트 시크릿 정보를 구성 맵에 저장했습니다. 이번 업데이트를 통해 Operator는 이 정보를 시크릿에 저장합니다. ([LOG-3717](#))
- 이번 업데이트 이전에는 Fluentd 수집기에서 **/var/log/auth-server/audit.log** 에 저장된 OAuth 로그인 이벤트를 캡처하지 않았습니다. 이번 업데이트를 통해 Fluentd는 이러한 OAuth 로그인 이벤트를 캡처하여 문제를 해결합니다. ([LOG-3729](#))

1.3.12.2. CVE

- [CVE-2020-10735](#)
- [CVE-2021-28861](#)
- [CVE-2022-2873](#)
- [CVE-2022-4415](#)
- [CVE-2022-40897](#)
- [CVE-2022-41222](#)
- [CVE-2022-43945](#)
- [CVE-2022-45061](#)
- [CVE-2022-48303](#)

1.3.13. Logging 5.6.2

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.6.2](#) 가 포함되어 있습니다.

1.3.13.1. 버그 수정

- 이번 업데이트 이전에는 systemd 로그의 우선 순위에 따라 수집기에서 수준 필드를 올바르게 설정하지 않았습니다. 이번 업데이트를 통해 수준 필드가 올바르게 설정됩니다. ([LOG-3429](#))
- 이번 업데이트 이전에는 Operator가 OpenShift Container Platform 4.12 이상에서 비호환 경고가 잘못 생성되었습니다. 이번 업데이트를 통해 Operator max OpenShift Container Platform 버전 값이 수정되어 문제를 해결합니다. ([LOG-3584](#))

- 이번 업데이트 이전에는 output 값이 **default**인 **ClusterLogForwarder** CR(사용자 정의 리소스)을 생성하면 오류가 발생하지 않았습니다. 이번 업데이트를 통해 이 값이 유효하지 않음을 알리는 오류 경고가 적절하게 생성됩니다. ([LOG-3437](#))
- 이번 업데이트 이전에는 **ClusterLogForwarder** CR(사용자 정의 리소스)에 하나의 출력이 기본으로 설정된 여러 파이프라인이 구성된 경우 수집기 Pod가 다시 시작됩니다. 이번 업데이트를 통해 출력 검증 논리가 수정되어 문제를 해결합니다. ([LOG-3559](#))
- 이번 업데이트 이전에는 생성 후 수집기 Pod가 다시 시작됩니다. 이번 업데이트를 통해 배포된 수집기가 자체적으로 다시 시작되지 않습니다. ([LOG-3608](#))
- 이번 업데이트 이전에는 패치 릴리스가 카탈로그에서 이전 버전의 Operator를 제거했습니다. 이로 인해 이전 버전을 설치할 수 없습니다. 이번 업데이트에서는 동일한 마이너 버전의 이전 릴리스가 카탈로그에 남아 있도록 번들 구성을 변경합니다. ([LOG-3635](#))

1.3.13.2. CVE

- [CVE-2022-23521](#)
- [CVE-2022-40303](#)
- [CVE-2022-40304](#)
- [CVE-2022-41903](#)
- [CVE-2022-47629](#)
- [CVE-2023-21835](#)
- [CVE-2023-21843](#)

1.3.14. Logging 5.6.1

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.6.1](#) 이 포함되어 있습니다.

1.3.14.1. 버그 수정

- 이번 업데이트 이전에는 보존이 활성 상태일 때 querier와의 통신에서 TLS 인증서 오류를 보고합니다. 이번 업데이트를 통해 compactor 및 querier는 더 이상 HTTP를 통해 잘못 통신하지 않습니다. ([LOG-3494](#))
- 이번 업데이트 이전에는 로키스 Operator에서 **LokiStack** CR의 상태 설정을 재시도하지 않아 오래된 상태 정보가 발생했습니다. 이번 업데이트를 통해 Operator는 충돌 시 상태 정보 업데이트를 다시 시도합니다. ([LOG-3496](#))
- 이번 업데이트 이전에는 **kube-apiserver-operator** Operator가 Webhook 유효성을 검사할 때 Loki Operator Webhook 서버에서 TLS 오류가 발생했습니다. 이번 업데이트를 통해 Loki Operator Webhook PKI는 OLM(Operator Lifecycle Manager)에서 관리하며 문제를 해결합니다. ([LOG-3510](#))
- 이번 업데이트 이전에는 LokiStack Gateway Labels Enforcer에서 부울 표현식과 결합된 라벨 필터를 사용할 때 유효한 LogQL 쿼리에 대한 구문 분석 오류가 발생했습니다. 이번 업데이트를 통해 LokiStack LogQL 구현에서는 부울 표현식이 있는 라벨 필터를 지원하고 문제를 해결합니다. ([LOG-3441](#)), ([LOG-3397](#))

- 이번 업데이트 이전에는 여러 라벨 키에 동일한 접두사와 일부 키가 포함된 경우 Elasticsearch에 작성된 레코드가 실패했습니다. 이번 업데이트를 통해 밑줄이 레이블 키의 점을 교체하여 문제를 해결합니다. ([LOG-3463](#))
- 이번 업데이트 이전에는 OpenShift Container Platform 콘솔과 logging-view-plugin 간의 비호환성으로 인해 OpenShift Container Platform 4.10 클러스터에 **Red Hat OpenShift Logging Operator**를 사용할 수 없었습니다. 이번 업데이트를 통해 플러그인이 OpenShift Container Platform 4.10 관리 콘솔과 올바르게 통합됩니다. ([LOG-3447](#))
- 이 업데이트 이전에는 **ClusterLogForwarder** 사용자 정의 리소스의 조정이 기본 로그 저장소를 참조하는 파이프라인의 성능이 저하된 상태를 잘못 보고했습니다. 이번 업데이트를 통해 파이프라인이 올바르게 검증됩니다. ([LOG-3477](#))

1.3.14.2. CVE

- [CVE-2021-46848](#)
- [CVE-2022-3821](#)
- [CVE-2022-35737](#)
- [CVE-2022-42010](#)
- [CVE-2022-42011](#)
- [CVE-2022-42012](#)
- [CVE-2022-42898](#)
- [CVE-2022-43680](#)
- [CVE-2021-35065](#)
- [CVE-2022-46175](#)

1.3.15. Logging 5.6.0

이 릴리스에는 [OpenShift Logging 릴리스 5.6](#) 이 포함되어 있습니다.

1.3.15.1. 사용 중단 알림

버전 5.6에서 Fluentd는 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다. Red Hat은 현재 릴리스 라이프사이클 동안 이 기능에 대한 버그 수정 및 지원을 제공하지만 이 기능은 더 이상 개선 사항을 제공하지 않으며 제거됩니다. Fluentd 대신 Vector를 사용할 수 있습니다.

1.3.15.2. 기능 개선

- 이번 업데이트를 통해 로깅은 OpenShift Container Platform 클러스터 전체 암호화 정책을 준수 합니다. ([로그-895](#))
- 이번 업데이트를 통해 RokiStack 사용자 정의 리소스를 통해 테넌트별, 스트림별, 글로벌 정책 보존 정책을 우선 순위에 따라 선언할 수 있습니다. ([LOG-2695](#))
- 이번 업데이트를 통해 Splunk는 로그 전달에 사용할 수 있는 출력 옵션입니다. ([LOG-2913](#))
- 이번 업데이트를 통해 Vector가 Fluentd를 기본 수집기로 대체합니다. ([로그-2222](#))

- 이번 업데이트를 통해 **Developer** 역 할은 OpenShift Container Platform 4.11 이상을 실행하는 클러스터의 로그 콘솔 플러그인 내에서 할당된 프로젝트별 워크로드 로그에 액세스할 수 있습니다. ([LOG-3388](#))
- 이번 업데이트를 통해 모든 소스의 로그에 Operator가 배포된 클러스터의 고유 식별자인 **openshift.cluster_id** 필드가 포함됩니다. 다음 명령을 사용하여 **clusterID** 값을 볼 수 있습니다.

```
$ oc get clusterversion/version -o jsonpath='{.spec.clusterID}{"\n"}'
```

([LOG-2715](#))

1.3.15.3. 확인된 문제

- 이번 업데이트 이전에는 여러 라벨 키에 동일한 접두사와 일부 키가 포함된 경우 Elasticsearch에서 로그를 거부합니다. 이로 인해 레이블 키의 를 `_`로 교체하여 Elasticsearch의 제한이 수정되었습니다. 이 문제를 해결하려면 오류를 유발하는 라벨을 제거하거나 라벨에 네임스페이스를 추가합니다. ([LOG-3463](#))

1.3.15.4. 버그 수정

- 이번 업데이트 이전에는 Kibana 사용자 정의 리소스를 삭제한 경우 OpenShift Container Platform 웹 콘솔에 Kibana에 대한 링크가 계속 표시됩니다. 이번 업데이트를 통해 Kibana 사용자 정의 리소스를 제거하면 해당 링크도 제거됩니다. ([LOG-2993](#))
- 이번 업데이트 이전에는 사용자가 액세스할 수 있는 네임스페이스의 애플리케이션 로그를 볼 수 없었습니다. 이번 업데이트를 통해 Loki Operator는 클러스터 역할 및 클러스터 역할 바인딩을 자동으로 생성하여 사용자가 애플리케이션 로그를 읽을 수 있습니다. ([로그-3072](#))
- 이번 업데이트 이전에는 Operator에서 LokiStack을 기본 로그 스토리지로 사용할 때 **ClusterLogForwarder** 사용자 정의 리소스에 정의된 모든 사용자 정의 출력을 제거했습니다. 이번 업데이트를 통해 Operator는 **ClusterLogForwarder** 사용자 정의 리소스를 처리할 때 기본 출력과 사용자 정의 출력을 병합합니다. ([로그-3090](#))
- 이번 업데이트 이전에는 CA 키를 로키에 마운트하기 위해 볼륨 이름으로 사용되었으며 CA 키에 접두과 같은 비 조정 문자가 포함된 경우 오류 상태가 발생했습니다. 이번 업데이트를 통해 볼륨 이름이 문제를 해결하는 내부 문자열로 표준화됩니다. ([LOG-3331](#))
- 이번 업데이트 이전에는 LokiStack 사용자 정의 리소스 정의에 설정된 기본값으로 인해 **Replication** ECDHE가 1인 1이 없이 로키스Stack 인스턴스를 생성할 수 없었습니다. 이번 업데이트를 통해 Operator는 사용된 크기의 실제 값을 설정합니다. ([LOG-3296](#))
- 이번 업데이트 이전에는 **structuredTypeKey** 또는 **structuredTypeName** 값을 정의하지 않고 JSON 구문 분석이 활성화된 경우 Vector에서 message 필드를 구문 분석했습니다. 이번 업데이트를 통해 Elasticsearch에 구조화된 로그를 작성할 때 **structuredTypeKey** 또는 **structuredTypeName**에 값이 필요합니다. ([LOG-3195](#))
- 이번 업데이트 이전에는 Elasticsearch Operator의 보안 생성 구성 요소가 내부 보안을 지속적으로 수정했습니다. 이번 업데이트를 통해 기존 보안이 올바르게 처리됩니다. ([LOG-3161](#))
- 이번 업데이트 이전에는 Elasticsearch 또는 Kibana 배포에 상태가 변경된 동안 Operator에서 제거 및 수집기 데몬 세트를 다시 생성하는 루프를 입력할 수 있었습니다. 이번 업데이트를 통해 Operator 상태 처리의 수정으로 문제가 해결됩니다. ([LOG-3157](#))
- 이번 업데이트 이전에는 Kibana에 **24h** OAuth 쿠키 만료 시간이 고정되어 있었기 때문에 **accessTokenInactivityTimeout** 필드가 **24h** 보다 낮은 값으로 설정될 때마다 Kibana에서 401 오

류가 발생했습니다. 이번 업데이트를 통해 Kibana의 OAuth 쿠키 만료 시간이 기본값인 **24h** 값을 사용하여 **accessTokenInactivityTimeout** 과 동기화됩니다. ([LOG-3129](#))

- 이번 업데이트 이전에는 리소스 조정을 위한 Operator 일반 패턴이 생성 후 지속적인 HTTP 409 응답을 가져오기 또는 업데이트하기 전에 생성을 시도하는 것이었습니다. 이번 업데이트를 통해 Operator는 먼저 오브젝트를 검색하고 오브젝트가 누락되었거나 지정되지 않은 경우에만 오브젝트를 생성하거나 업데이트합니다. ([LOG-2919](#))
- 이번 업데이트 이전에는 Fluentd의 **.level** 및 '**structure.level**' 필드에 다른 값이 포함될 수 있었습니다. 이번 업데이트를 통해 같은 각 필드에 대해 동일합니다. ([LOG-2819](#))
- 이번 업데이트 이전에는 Operator에서 신뢰할 수 있는 CA 번들 채우기를 기다리지 않고 번들이 업데이트되면 수집기를 두 번째로 배포했습니다. 이번 업데이트를 통해 Operator는 번들이 수집기 배포를 계속하기 전에 채워졌는지 여부를 간략하게 기다립니다. ([LOG-2789](#))
- 이번 업데이트 이전에는 지표를 검토할 때 로깅 Telemetry 정보가 두 번 표시되었습니다. 이번 업데이트를 통해 로깅 Telemetry 정보가 예상대로 표시됩니다. ([로그-2315](#))
- 이번 업데이트 이전에는 JSON 구문 분석 추가를 활성화한 후 Fluentd Pod 로그에 경고 메시지가 표시되었습니다. 이번 업데이트를 통해 해당 경고 메시지가 표시되지 않습니다. ([로그-1806](#))
- 이번 업데이트 이전에는 **oc**에 캐시를 빌드하는 데 쓰기 권한이 있는 폴더가 필요하므로 **must-gather** 스크립트가 완료되지 않았습니다. 이번 업데이트를 통해 **oc**는 폴더에 대한 쓰기 권한이 있고 **must-gather** 스크립트가 성공적으로 완료됩니다. ([LOG-3446](#))
- 이번 업데이트 이전에는 로그 수집기 SCC를 클러스터의 다른 SCC로 대체하여 컬렉터를 사용할 수 없게 됩니다. 이번 업데이트에서는 로그 수집기 SCC의 우선 순위를 설정하여 다른 것보다 우선합니다. ([LOG-3235](#))
- 이번 업데이트 이전에는 벡터가 필드 **시퀀스** 가 누락되었으며 실제 나노초 부족을 처리하는 방법으로 fluentd에 추가되었습니다. 이번 업데이트를 통해 **openshift.sequence** 필드가 이벤트 로그에 추가되었습니다. ([LOG-3106](#))

1.3.15.5. CVE

- [CVE-2020-36518](#)
- [CVE-2021-46848](#)
- [CVE-2022-2879](#)
- [CVE-2022-2880](#)
- [CVE-2022-27664](#)
- [CVE-2022-32190](#)
- [CVE-2022-35737](#)
- [CVE-2022-37601](#)
- [CVE-2022-41715](#)
- [CVE-2022-42003](#)
- [CVE-2022-42004](#)

- CVE-2022-42010
- CVE-2022-42011
- CVE-2022-42012
- CVE-2022-42898
- CVE-2022-43680

1.4. LOGGING 5.5



참고

로깅은 핵심 OpenShift Container Platform과 별도의 릴리스 주기와 함께 설치 가능한 구성 요소로 제공됩니다. [Red Hat OpenShift Container Platform 라이프 사이클 정책](#)은 릴리스 호환성에 대해 간단히 설명합니다.

1.4.1. Logging 5.5.18

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.5.18](#) 이 포함되어 있습니다.

1.4.1.1. 버그 수정

없음.

1.4.1.2. CVE

- CVE-2023-40217
- CVE-2023-44487

1.4.2. 로깅 5.5.17

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.5.17](#) 이 포함되어 있습니다.

1.4.2.1. 버그 수정

- 이번 업데이트 이전에는 이벤트 라우터에서 사용되지 않는 메트릭으로 인해 과도한 메모리 사용량으로 인해 컨테이너가 실패했습니다. 이번 업데이트를 통해 사용되지 않은 메트릭을 제거하여 이벤트 라우터의 메모리 사용량이 줄어듭니다. ([LOG-4688](#))

1.4.2.2. CVE

- CVE-2023-0800
- CVE-2023-0801
- CVE-2023-0802
- CVE-2023-0803
- CVE-2023-0804

- CVE-2023-2002
- CVE-2023-3090
- CVE-2023-3341
- CVE-2023-3390
- CVE-2023-3776
- CVE-2023-4004
- CVE-2023-4527
- CVE-2023-4806
- CVE-2023-4813
- CVE-2023-4863
- CVE-2023-4911
- CVE-2023-5129
- CVE-2023-20593
- CVE-2023-29491
- CVE-2023-30630
- CVE-2023-35001
- CVE-2023-35788

1.4.3. 로깅 5.5.16

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.5.16 이 포함되어 있습니다.

1.4.3.1. 버그 수정

- 이번 업데이트 이전에는 LokiStack 게이트웨이가 인증된 요청을 매우 광범위하게 캐시했습니다. 이로 인해 잘못된 권한 부여 결과가 발생했습니다. 이번 업데이트를 통해 LokiStack 게이트웨이 캐시는 이 문제를 보다 세분화하여 해결합니다. ([LOG-4434](#))

1.4.3.2. CVE

- CVE-2023-3899
- CVE-2023-32360
- CVE-2023-34969

1.4.4. 로깅 5.5.14

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.5.14 가 포함되어 있습니다.

1.4.4.1. 버그 수정

- 이번 업데이트 이전에는 로그에서 Vector 수집기에서 다음과 같은 오류 메시지를 패닉시킬 수 있습니다. '모든 분기에서 'vector-worker' panicked가 비활성화되고 다른 분기', [src/kubernetes/reflector.rs:26:9](#) 입니다. 이번 업데이트를 통해 벡터 수집기에 오류가 표시되지 않습니다. ([LOG-4279](#))

1.4.4.2. CVE

- [CVE-2023-2828](#)

1.4.5. 로깅 5.5.13

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.5.13](#) 이 포함되어 있습니다.

1.4.5.1. 버그 수정

없음.

1.4.5.2. CVE

- [CVE-2023-1999](#)
- [CVE-2020-24736](#)
- [CVE-2022-48281](#)
- [CVE-2023-1667](#)
- [CVE-2023-2283](#)
- [CVE-2023-24329](#)
- [CVE-2023-26604](#)
- [CVE-2023-28466](#)

1.4.6. 로깅 5.5.12

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.5.12](#) 가 포함되어 있습니다.

1.4.6.1. 버그 수정

없음.

1.4.6.2. CVE

- [CVE-2021-26341](#)
- [CVE-2021-33655](#)
- [CVE-2021-33656](#)
- [CVE-2022-1462](#)

- CVE-2022-1679
- CVE-2022-1789
- CVE-2022-2196
- CVE-2022-2663
- CVE-2022-3028
- CVE-2022-3239
- CVE-2022-3522
- CVE-2022-3524
- CVE-2022-3564
- CVE-2022-3566
- CVE-2022-3567
- CVE-2022-3619
- CVE-2022-3623
- CVE-2022-3625
- CVE-2022-3627
- CVE-2022-3628
- CVE-2022-3707
- CVE-2022-3970
- CVE-2022-4129
- CVE-2022-20141
- CVE-2022-25147
- CVE-2022-25265
- CVE-2022-30594
- CVE-2022-35252
- CVE-2022-36227
- CVE-2022-39188
- CVE-2022-39189
- CVE-2022-41218
- CVE-2022-41674

- CVE-2022-42703

- CVE-2022-42720

- CVE-2022-42721

- CVE-2022-42722

- CVE-2022-43552

- CVE-2022-43750

- CVE-2022-47929

- CVE-2023-0394

- CVE-2023-0461

- CVE-2023-1195

- CVE-2023-1582

- CVE-2023-2491

- CVE-2023-22490

- CVE-2023-23454

- CVE-2023-23946

- CVE-2023-25652

- CVE-2023-25815

- CVE-2023-27535

- CVE-2023-29007

1.4.7. 로깅 5.5.11

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.5.11 이 포함되어 있습니다.

1.4.7.1. 버그 수정

- 이번 업데이트 이전에는 로그 히스토그램을 클릭하고 끌어서 OpenShift Container Platform 웹 콘솔에서 시간 범위를 선택할 수 없었습니다. 이번 업데이트를 통해 클릭 및 드래그를 사용하여 시간 범위를 성공적으로 선택할 수 있습니다. ([LOG-4102](#))
- 이번 업데이트 이전에는 OpenShift Container Platform 웹 콘솔의 리소스 표시 링크를 클릭하면 적용되지 않았습니다. 이번 업데이트를 통해 리소스 표시 링크의 기능을 수정하여 각 로그 항목에 대한 리소스 표시를 전환하여 문제가 해결됩니다. ([LOG-4117](#))

1.4.7.2. CVE

- CVE-2021-26341

- CVE-2021-33655

- CVE-2021-33656
- CVE-2022-1462
- CVE-2022-1679
- CVE-2022-1789
- CVE-2022-2196
- CVE-2022-2663
- CVE-2022-2795
- CVE-2022-3028
- CVE-2022-3239
- CVE-2022-3522
- CVE-2022-3524
- CVE-2022-3564
- CVE-2022-3566
- CVE-2022-3567
- CVE-2022-3619
- CVE-2022-3623
- CVE-2022-3625
- CVE-2022-3627
- CVE-2022-3628
- CVE-2022-3707
- CVE-2022-3970
- CVE-2022-4129
- CVE-2022-20141
- CVE-2022-24765
- CVE-2022-25265
- CVE-2022-29187
- CVE-2022-30594
- CVE-2022-36227
- CVE-2022-39188

- CVE-2022-39189
- CVE-2022-39253
- CVE-2022-39260
- CVE-2022-41218
- CVE-2022-41674
- CVE-2022-42703
- CVE-2022-42720
- CVE-2022-42721
- CVE-2022-42722
- CVE-2022-43750
- CVE-2022-47929
- CVE-2023-0394
- CVE-2023-0461
- CVE-2023-1195
- CVE-2023-1582
- CVE-2023-2491
- CVE-2023-23454
- CVE-2023-27535

1.4.8. Logging 5.5.10

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.5.10 이 포함되어 있습니다.

1.4.8.1. 버그 수정

- 이번 업데이트 이전에는 OpenShift 웹 콘솔의 로깅 보기 플러그인에 연결할 수 없는 경우 오류 텍스트만 표시되었습니다. 이번 업데이트 후 플러그인에 연결할 수 없는 stack을 수정하는 방법에 대한 세부 사항이 포함된 적절한 오류 메시지가 표시됩니다. ([LOG-2874](#))

1.4.8.2. CVE

- CVE-2022-4304
- CVE-2022-4450
- CVE-2023-0215
- CVE-2023-0286
- CVE-2023-0361

- CVE-2023-23916

1.4.9. Logging 5.5.9

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.5.9 가 포함되어 있습니다.

1.4.9.1. 버그 수정

- 이번 업데이트 이전에는 Fluentd 수집기의 문제로 인해 **/var/log/auth-server/audit.log**에 저장된 OAuth 로그인 이벤트가 캡처되지 않았습니다. 이로 인해 OAuth 서비스에서 로그인 이벤트가 불완전해졌습니다. 이번 업데이트를 통해 Fluentd 수집기는 이제 예상대로 **/var/log/auth-server/audit.log**에 저장된 항목을 포함하여 OAuth 서비스에서 모든 로그인 이벤트를 캡처하여 이 문제를 해결합니다.<https://issues.redhat.com/browse/LOG-3730>
- 이번 업데이트 이전에는 구조화된 구문 분석이 활성화되고 메시지가 여러 대상으로 전달되면 깊은 복사되지 않았습니다. 이로 인해 구조화된 메시지를 포함하여 수신된 로그 중 일부가 생성되었지만 다른 로그는 그렇지 않았습니다. 이번 업데이트를 통해 JSON 구문 분석 전에 구성 생성이 깊은 복사 메시지로 수정되었습니다. 결과적으로 여러 대상으로 전달되는 경우에도 수신된 모든 로그에 구조화된 메시지가 포함됩니다.([LOG-3767](#)).

1.4.9.2. CVE

- CVE-2022-4304
- CVE-2022-4450
- CVE-2022-41717
- CVE-2023-0215
- CVE-2023-0286
- CVE-2023-0767
- CVE-2023-23916

1.4.10. Logging 5.5.8

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.5.8 이 포함되어 있습니다.

1.4.10.1. 버그 수정

- 이번 업데이트 이전에는 수집기가 수준 필드를 설정하는 방법에 오류가 있기 때문에 **systemd**로 그에서 **priority** 필드가 누락되었습니다. 이번 업데이트를 통해 이러한 필드가 올바르게 설정되어 문제를 해결합니다. ([LOG-3630](#))

1.4.10.2. CVE

- CVE-2020-10735
- CVE-2021-28861
- CVE-2022-2873
- CVE-2022-4415

- CVE-2022-24999
- CVE-2022-40897
- CVE-2022-41222
- CVE-2022-41717
- CVE-2022-43945
- CVE-2022-45061
- CVE-2022-48303

1.4.11. Logging 5.5.7

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.5.7](#) 이 포함되어 있습니다.

1.4.11.1. 버그 수정

- 이번 업데이트 이전에는 LokiStack Gateway Labels Enforcer에서 부울 표현식과 결합된 라벨 필터를 사용할 때 유효한 LogQL 쿼리에 대한 구문 분석 오류가 발생했습니다. 이번 업데이트를 통해 LokiStack LogQL 구현에서는 부울 표현식이 있는 라벨 필터를 지원하고 문제를 해결합니다. ([LOG-3534](#))
- 이번 업데이트 이전에는 **ClusterLogForwarder** CR(사용자 정의 리소스)에서 syslog 출력의 TLS 인증 정보를 Fluentd에 전달하지 않아 전송 중에 오류가 발생했습니다. 이번 업데이트를 통해 인증 정보가 Fluentd에 올바르게 전달되어 문제를 해결합니다. ([LOG-3533](#))

1.4.11.2. CVE

[CVE-2021-46848](#) [CVE-2022-3821](#) [CVE-2022-35737](#) [CVE-2022-42010](#) [CVE-2022-42011](#) [CVE-2022-42898](#) [CVE-2022-43680](#)

1.4.12. 로깅 5.5.6

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.5.6](#) 이 포함되어 있습니다.

1.4.12.1. 버그 수정

- 이번 업데이트 이전에는 Pod 보안 승인 컨트롤러가 **openshift-logging** 네임스페이스에 라벨 **podSecurityLabelSync = true** 를 추가했습니다. 이로 인해 지정된 보안 레이블을 덮어쓰고 결과적으로 Collector Pod가 시작되지 않았습니다. 이번 업데이트를 통해 **podSecurityLabelSync = false** 레이블은 보안 라벨을 유지합니다. 수집기 Pod는 예상대로 배포됩니다. ([LOG-3340](#))
- 이번 업데이트 이전에는 Operator가 클러스터에서 활성화되지 않은 경우에도 콘솔 보기 플러그인을 설치했습니다. 이로 인해 Operator가 충돌했습니다. 이번 업데이트를 통해 클러스터의 계정에 콘솔 보기 기능이 활성화되어 있지 않으면 Operator가 정상적으로 작동하고 콘솔 보기 기능을 설치하지 않습니다. ([LOG-3407](#))
- 이번 업데이트 이전에는 Elasticsearch 배포 상태를 업데이트하지 않는 경우 **Red Hat Elasticsearch Operator** 가 배포되지 않은 경우 Operator가 충돌하는 희귀를 지원하기 위한 사전 설정입니다. 이번 업데이트를 통해 해당 설정 사항이 복원되어 Operator가 안정적이지만 보고된 상태와 관련된 이전 문제가 다시 도입되었습니다. ([LOG-3428](#))

- 이번 업데이트 이전에는 Loki Operator가 선택한 스택 크기에 관계없이 LokiStack 게이트웨이의 복제본 하나를 배포했습니다. 이번 업데이트를 통해 선택한 크기에 따라 복제본 수가 올바르게 구성됩니다. ([로그-3478](#))
- 이번 업데이트 이전에는 여러 라벨 키에 동일한 접두사와 일부 키가 포함된 경우 Elasticsearch에 작성된 레코드가 실패했습니다. 이번 업데이트를 통해 밑줄이 레이블 키의 점을 교체하여 문제를 해결합니다. ([LOG-3341](#))
- 이번 업데이트 이전에는 로깅 보기 플러그인에 특정 버전의 OpenShift Container Platform에 대한 호환되지 않는 기능이 포함되어 있었습니다. 이번 업데이트를 통해 플러그인의 올바른 릴리스 스트림이 문제를 해결합니다. ([로그-3467](#))
- 이번 업데이트 이전에는 **ClusterLogForwarder** 사용자 정의 리소스의 조정이 하나 이상의 파일 프라인의 성능 저하 상태를 잘못 보고하여 수집기 Pod가 8-10초마다 다시 시작됩니다. 이번 업데이트를 통해 **ClusterLogForwarder** 사용자 정의 리소스 프로세스를 올바르게 조정하여 문제를 해결합니다. ([LOG-3469](#))
- 이를 변경하기 전에 ClusterLogForwarder 사용자 정의 리소스의 **outputDefaults** 필드에 대한 사양이 선언된 모든 Elasticsearch 출력 유형에 설정을 적용합니다. 이 변경으로 인해 설정이 기본 관리 Elasticsearch 저장소에 구체적으로 적용되는 개선 사양과 일치하도록 동작이 수정되었습니다. ([LOG-3342](#))
- 이번 업데이트 이전에는 OpenShift CLI(oc)에 캐시를 빌드할 수 있는 쓰기 권한이 있는 폴더가 필요하므로 OpenShift CLI(oc) **must-gather** 스크립트가 완료되지 않았습니다. 이번 업데이트를 통해 OpenShift CLI(oc)에 폴더에 대한 쓰기 권한이 있으며 **must-gather** 스크립트가 성공적으로 완료됩니다. ([LOG-3472](#))
- 이번 업데이트 이전에는 Loki Operator 웹 후크 서버에서 TLS 오류가 발생했습니다. 이번 업데이트를 통해 Loki Operator webhook PKI는 Operator Lifecycle Manager의 동적 웹 후크 관리에서 이 문제를 해결합니다. ([LOG-3511](#))

1.4.12.2. CVE

- [CVE-2021-46848](#)
- [CVE-2022-2056](#)
- [CVE-2022-2057](#)
- [CVE-2022-2058](#)
- [CVE-2022-2519](#)
- [CVE-2022-2520](#)
- [CVE-2022-2521](#)
- [CVE-2022-2867](#)
- [CVE-2022-2868](#)
- [CVE-2022-2869](#)
- [CVE-2022-2953](#)
- [CVE-2022-2964](#)

- [CVE-2022-4139](#)
- [CVE-2022-35737](#)
- [CVE-2022-42010](#)
- [CVE-2022-42011](#)
- [CVE-2022-42012](#)
- [CVE-2022-42898](#)
- [CVE-2022-43680](#)

1.4.13. 로깅 5.5.5

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.5.5](#) 가 포함되어 있습니다.

1.4.13.1. 버그 수정

- 이번 업데이트 이전에는 Kibana에 **24h** OAuth 쿠키 만료 시간이 고정되어 있었기 때문에 **accessTokenInactivityTimeout** 필드가 **24h** 보다 낮은 값으로 설정될 때마다 Kibana에서 401 오류가 발생했습니다. 이번 업데이트를 통해 Kibana의 OAuth 쿠키 만료 시간이 기본값인 **24h** 값을 사용하여 **accessTokenInactivityTimeout** 과 동기화됩니다. ([LOG-3305](#))
- 이번 업데이트 이전에는 **structuredTypeKey** 또는 **structuredTypeName** 값을 정의하지 않고 JSON 구문 분석이 활성화된 경우 Vector에서 message 필드를 구문 분석했습니다. 이번 업데이트를 통해 Elasticsearch에 구조화된 로그를 작성할 때 **structuredTypeKey** 또는 **structuredTypeName**에 값이 필요합니다. ([LOG-3284](#))
- 이번 업데이트 이전에는 이 경고 표현식에서 반환된 라벨 세트에 카디널리티 문제가 있을 때 **FluentdQueueLengthIncreasing** 경고가 실행되지 않을 수 있었습니다. 이번 업데이트에서는 경고에 필요한 레이블만 포함하도록 라벨을 줄입니다. ([LOG-3226](#))
- 이번 업데이트 이전에는 로키가 연결이 끊긴 클러스터의 외부 스토리지에 도달할 수 있는 기능을 지원하지 않았습니다. 이번 업데이트를 통해 프록시 환경 변수 및 프록시 신뢰할 수 있는 CA 번들 이 컨테이너 이미지에 포함되어 이러한 연결을 지원합니다. ([LOG-2860](#))
- 이번 업데이트 이전에는 OpenShift Container Platform 웹 콘솔 사용자가 로키에 대한 CA 인증서가 포함된 **ConfigMap** 오브젝트를 선택할 수 없어 CA 없이 Pod가 작동할 수 없었습니다. 이번 업데이트를 통해 웹 콘솔 사용자는 구성 맵을 선택하여 문제를 해결할 수 있습니다. ([LOG-3310](#))
- 이번 업데이트 이전에는 CA 키를 로키에 마운트하기 위해 볼륨 이름으로 사용되었으며 CA 키에 비 조정 문자(예: dots)가 포함된 경우 오류 상태가 발생했습니다. 이번 업데이트를 통해 볼륨 이름이 문제를 해결하는 내부 문자열로 표준화됩니다. ([LOG-3332](#))

1.4.13.2. CVE

- [CVE-2016-3709](#)
- [CVE-2020-35525](#)
- [CVE-2020-35527](#)
- [CVE-2020-36516](#)

- CVE-2020-36558
- CVE-2021-3640
- CVE-2021-30002
- CVE-2022-0168
- CVE-2022-0561
- CVE-2022-0562
- CVE-2022-0617
- CVE-2022-0854
- CVE-2022-0865
- CVE-2022-0891
- CVE-2022-0908
- CVE-2022-0909
- CVE-2022-0924
- CVE-2022-1016
- CVE-2022-1048
- CVE-2022-1055
- CVE-2022-1184
- CVE-2022-1292
- CVE-2022-1304
- CVE-2022-1355
- CVE-2022-1586
- CVE-2022-1785
- CVE-2022-1852
- CVE-2022-1897
- CVE-2022-1927
- CVE-2022-2068
- CVE-2022-2078
- CVE-2022-2097
- CVE-2022-2509

- CVE-2022-2586
- CVE-2022-2639
- CVE-2022-2938
- CVE-2022-3515
- CVE-2022-20368
- CVE-2022-21499
- CVE-2022-21618
- CVE-2022-21619
- CVE-2022-21624
- CVE-2022-21626
- CVE-2022-21628
- CVE-2022-22624
- CVE-2022-22628
- CVE-2022-22629
- CVE-2022-22662
- CVE-2022-22844
- CVE-2022-23960
- CVE-2022-24448
- CVE-2022-25255
- CVE-2022-26373
- CVE-2022-26700
- CVE-2022-26709
- CVE-2022-26710
- CVE-2022-26716
- CVE-2022-26717
- CVE-2022-26719
- CVE-2022-27404
- CVE-2022-27405
- CVE-2022-27406

- CVE-2022-27950
- CVE-2022-28390
- CVE-2022-28893
- CVE-2022-29581
- CVE-2022-30293
- CVE-2022-34903
- CVE-2022-36946
- CVE-2022-37434
- CVE-2022-39399

1.4.14. 로깅 5.5.4

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.5.4 가 포함되어 있습니다.

1.4.14.1. 버그 수정

- 이번 업데이트 이전에는 로깅 보기 플러그인의 쿼리 구문 분석기에 있는 오류로 인해 쿼리에 중괄호 {} 가 포함되어 있는 경우 로그 쿼리의 일부가 사라졌습니다. 이로 인해 쿼리가 유효하지 않아 유효한 쿼리에 대해 오류가 반환되었습니다. 이번 업데이트를 통해 구문 분석기는 이러한 쿼리를 올바르게 처리합니다. ([로그-3042](#))
- 이번 업데이트 이전에는 Elasticsearch 또는 Kibana 배포에 상태가 변경된 동안 Operator에서 제거 및 수집기 데몬 세트를 다시 생성하는 루프를 입력할 수 있었습니다. 이번 업데이트를 통해 Operator 상태 처리의 수정으로 문제가 해결됩니다. ([로그-3049](#))
- 이번 업데이트 이전에는 Vector의 수집기 구현을 지원하기 위해 경고가 구현되지 않았습니다. 이 변경으로 벡터 경고가 추가되고 선택한 수집기 구현에 따라 별도의 경고를 배포합니다. ([LOG-3127](#))
- 이번 업데이트 이전에는 Elasticsearch Operator의 보안 생성 구성 요소가 내부 보안을 지속적으로 수정했습니다. 이번 업데이트를 통해 기존 보안이 올바르게 처리됩니다. ([LOG-3138](#))
- 이번 업데이트 이전에는 로깅 **must-gather** 스크립트를 사전 리팩터링하여 아티팩트에 대한 예상 위치를 제거했습니다. 이번 업데이트에서는 아티팩트를 /**must-gather** 폴더에 작성하도록 변경 사항을 되돌립니다. ([LOG-3213](#))
- 이번 업데이트 이전에는 특정 클러스터에서 Prometheus 내보내기를 IPv6 대신 IPv4에 바인딩했습니다. 이번 업데이트 후 Fluentd는 IP 버전을 감지하고 IPv4의 경우 **0.0.0.0** 또는 IPv6의 경우 [::] 에 바인딩합니다. ([LOG-3162](#))

1.4.14.2. CVE

- CVE-2020-35525
- CVE-2020-35527
- CVE-2022-0494

- CVE-2022-1353
- CVE-2022-2509
- CVE-2022-2588
- CVE-2022-3515
- CVE-2022-21618
- CVE-2022-21619
- CVE-2022-21624
- CVE-2022-21626
- CVE-2022-21628
- CVE-2022-23816
- CVE-2022-23825
- CVE-2022-29900
- CVE-2022-29901
- CVE-2022-32149
- CVE-2022-37434
- CVE-2022-40674

1.4.15. Logging 5.5.3

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.5.3](#) 이 포함되어 있습니다.

1.4.15.1. 버그 수정

- 이번 업데이트 이전에는 구조화된 메시지가 포함된 로그 항목에 원래 메시지 필드가 포함되어 있었기 때문에 항목이 더 커졌습니다. 이번 업데이트에서는 구조화된 로그의 message 필드를 제거하여 증가된 크기를 줄입니다. ([LOG-2759](#))
- 이번 업데이트 이전에는 수집기 구성이 수집기 **default-log-store** 및 **visualization** Pod에서 로그를 제외했지만 **.gz** 파일에 보관된 로그를 제외할 수 없었습니다. 이번 업데이트를 통해 수집기 **default-log-store** 및 **visualization** 포드의 **.gz** 파일로 저장된 아카이브 로그도 제외됩니다. ([LOG-2844](#))
- 이번 업데이트 이전에는 게이트웨이를 통해 사용할 수 없는 **pod**에 대한 요청이 전송되면 중단에 대한 경고 메시지가 표시되지 않습니다. 이번 업데이트를 통해 게이트웨이에 쓰기 또는 읽기 문제에 있는 경우 개별 경고가 생성됩니다. ([LOG-2884](#))
- 이번 업데이트 이전에는 파이프라인을 통해 참조를 통해 전달된 값 때문에 **fluent** 플러그인

에 의해 **pod** 메타데이터를 변경할 수 있었습니다. 이번 업데이트를 통해 각 로그 메시지가 개별적으로 처리할 수 있도록 **Pod** 메타데이터 복사본을 수신합니다. ([로그-3046](#))

- 이번 업데이트 이전에는 **OpenShift** 콘솔 로그에서 알 수 없는 심각도를 선택하면 **level=unknown** 값이 있는 제외된 로그를 확인합니다. 이번 업데이트를 통해 수준이 없고 **level=unknown** 값이 없는 로그가 알 수 없는 심각도로 필터링될 때 표시됩니다. ([로그-3062](#))
- 이번 업데이트 이전에는 **Elasticsearch**로 전송된 로그 레코드에 로그를 보낼 인덱스 이름이 포함된 **write-index**라는 추가 필드가 있었습니다. 이 필드는 데이터 모델의 일부가 아닙니다. 이번 업데이트 후에는 이 필드가 더 이상 전송되지 않습니다. ([로그-3075](#))
- 새로운 내장 **Pod 보안 승인** 컨트롤러가 도입됨에 따라 **Pod**는 전역 또는 네임스페이스 수준에서 정의된 적용된 보안 표준에 따라 구성되지 않습니다. 이번 업데이트를 통해 **Operator** 및 수집기는 권한 있는 실행을 허용하고 보안 감사 경고 또는 오류 없이 실행됩니다. ([로그-3077](#))
- 이번 업데이트 이전에는 **Operator**에서 **LokiStack**을 기본 로그 스토리지로 사용할 때 **ClusterLogForwarder** 사용자 정의 리소스에 정의된 모든 사용자 정의 출력을 제거했습니다. 이번 업데이트를 통해 **Operator**는 **ClusterLogForwarder** 사용자 정의 리소스를 처리할 때 기본 출력과 사용자 정의 출력을 병합합니다. ([로그-3095](#))

1.4.15.2. CVE

- [CVE-2015-20107](#)
- [CVE-2022-0391](#)
- [CVE-2022-2526](#)
- [CVE-2022-21123](#)
- [CVE-2022-21125](#)

- [CVE-2022-21166](#)
 - [CVE-2022-29154](#)
 - [CVE-2022-32206](#)
 - [CVE-2022-32208](#)
 - [CVE-2022-34903](#)
- #### 1.4.16. logging 5.5.2
- 이 릴리스에는 [OpenShift Logging](#) 버그 수정 릴리스 5.5.2가 포함되어 있습니다.
- ##### 1.4.16.1. 버그 수정
- 이번 업데이트 이전에는 **Fluentd** 수집기에 대한 경고 규칙이 **OpenShift Container Platform** 모니터링 스타일 지침을 준수하지 않았습니다. 이번 업데이트에서는 네임스페이스 레이블을 포함하도록 이러한 경고를 수정하여 문제를 해결합니다. ([LOG-1823](#))
 - 이번 업데이트 이전에는 인덱스 이름에 하이픈 문자가 두 개 이상 있을 때마다 인덱스 관리를 오버 스크립트가 새 인덱스 이름을 생성하지 못했습니다. 이번 업데이트를 통해 인덱스 이름이 올바르게 생성됩니다. ([LOG-2644](#))
 - 이번 업데이트 이전에는 **Kibana** 경로가 인증서가 없는 **caCertificate** 값을 설정했습니다. 이번 업데이트를 통해 **caCertificate** 값이 설정되지 않았습니다. ([LOG-2661](#))
 - 이번 업데이트 이전에는 수집기 종속성을 변경하면 사용되지 않는 매개변수에 대한 경고 메시지가 표시되었습니다. 이번 업데이트를 통해 사용하지 않는 설정 매개변수를 제거하면 문제가 해결됩니다. ([LOG-2859](#))
 - 이번 업데이트 이전에는 **Loki Operator**가 생성한 배포에 생성된 Pod가 Linux가 아닌 운영체제로 등록되는 문제를 해결했습니다. ([LOG-29154](#))

체제가 있는 노드에 실수로 예약되었습니다(Operator가 실행 중인 클러스터에서 이러한 노드를 사용할 수 있는 경우). 이번 업데이트를 통해 Operator는 Linux 기반 노드에서만 Pod를 예약할 수 있는 Pod 정의에 추가 노드 선택기를 연결합니다. ([LOG-2895](#))

- 이번 업데이트 이전에는 LokiStack 게이트웨이의 LogQL 구문 분석기 문제로 인해 OpenShift 콘솔 로그 보기와 함께 심각도별로 로그를 필터링하지 않았습니다. 이번 업데이트를 통해 구문 분석 수정으로 문제가 해결되고 OpenShift Console Logs 보기에서 심각도별로 필터링할 수 있습니다. ([LOG-2908](#))
- 이번 업데이트 이전에는 Fluentd 수집기 플러그인 리팩터링이 이벤트의 타임 스탬프 필드를 제거했습니다. 이번 업데이트에서는 이벤트의 수신된 시간에서 가져온 타임스탬프 필드를 복원합니다. ([LOG-2923](#))
- 이번 업데이트 이전에는 감사 로그에 수준 필드가 없으면 벡터 로그에서 오류가 발생했습니다. 이번 업데이트를 통해 감사 로그 레코드에 수준 필드를 추가하면 문제가 해결됩니다. ([LOG-2961](#))
- 이번 업데이트 이전에는 Kibana 사용자 정의 리소스를 삭제한 경우 OpenShift Container Platform 웹 콘솔에 Kibana에 대한 링크가 계속 표시됩니다. 이번 업데이트를 통해 Kibana 사용자 정의 리소스를 제거하면 해당 링크도 제거됩니다. ([LOG-3053](#))
- 이번 업데이트 이전에는 ClusterLogForwarder 사용자 정의 리소스에 JSON 구문 분석이 정의된 경우 각 롤오버 작업이 빈 인덱스를 생성했습니다. 이번 업데이트를 통해 새 인덱스가 비어 있지 않습니다. ([LOG-3063](#))
- 이번 업데이트 이전에는 Loki Operator 5.4에서 처음 생성한 Loki Operator 5.5 리소스가 업데이트 된 후 사용자가 LokiStack을 삭제한 경우 남아 있습니다. 이번 업데이트를 통해 리소스의 소유자가 5.5 LokiStack을 가리킵니다. ([LOG-2945](#))
- 이번 업데이트 이전에는 사용자가 액세스할 수 있는 네임스페이스의 애플리케이션 로그를 볼 수 없었습니다. 이번 업데이트를 통해 Loki Operator는 클러스터 역할 및 클러스터 역할 바인딩을 자동으로 생성하여 사용자가 애플리케이션 로그를 읽을 수 있습니다. ([LOG-2918](#))
- 이번 업데이트 이전에는 cluster-admin 권한이 있는 사용자가 로깅 콘솔을 사용하여 인프라 및 감사 로그를 올바르게 볼 수 없었습니다. 이번 업데이트를 통해 cluster-admin 및 dedicated-admin 그룹의 사용자도 admins로 인식하도록 권한 부여 검사가 확장되었습니다. ([LOG-2970](#))

1.4.16.2. CVE

- [CVE-2015-20107](#)
- [CVE-2022-0391](#)
- [CVE-2022-21123](#)
- [CVE-2022-21125](#)
- [CVE-2022-21166](#)
- [CVE-2022-29154](#)
- [CVE-2022-32206](#)
- [CVE-2022-32208](#)
- [CVE-2022-34903](#)

1.4.17. 5.5.1 로깅

이 릴리스에는 [OpenShift Logging 버그 수정 릴리스 5.5.1](#) 이 포함되어 있습니다.

1.4.17.1. 기능 개선

- 이번 개선된 기능에는 로깅 콘솔 플러그인을 사용할 때 OpenShift Container Platform 웹 콘솔의 Pod 세부 정보 페이지에 집계된 로그 탭이 추가되었습니다. 이 향상된 기능은 [OpenShift Container Platform 4.10](#) 이상에서만 사용할 수 있습니다. ([LOG-2647](#))
-

이번 개선된 기능에는 **Google Cloud Logging**이 로그 전달의 출력 옵션으로 추가되었습니다. ([LOG-1482](#))

1.4.17.2. 버그 수정

- 이번 업데이트 이전에는 **Operator**에서 **Pod**가 준비되었는지 확인하지 않아 클러스터를 다시 시작하는 동안 클러스터가 작동하지 않는 상태에 도달했습니다. 이번 업데이트를 통해 **Operator**는 재시작 중에 새 **Pod**를 계속 진행하기 전에 새 **Pod**를 준비 상태로 표시하여 문제를 해결합니다. ([LOG-2745](#))
- 이번 업데이트 이전에는 **Fluentd**가 **Kubernetes** 플랫폼이 로그 파일을 헤전시키고 더 이상 로그 메시지를 읽지 않음을 인식하지 못하는 경우가 있었습니다. 이번 업데이트에서는 업스트림 개발 팀에서 권장하는 구성 매개 변수를 설정하여 이를 수정합니다. ([LOG-2995](#))
- 이번 업데이트 이전에는 여러 줄 오류 감지 기능을 추가하여 내부 라우팅이 변경되어 레코드를 잘못된 대상으로 전달했습니다. 이번 업데이트를 통해 내부 라우팅이 잘못되었습니다. ([LOG-2801](#))
- 이번 업데이트 이전에는 **OpenShift Container Platform** 웹 콘솔의 새로 고침 간격을 변경하면 쿼리 필드가 비어 있을 때 오류가 발생했습니다. 이번 업데이트를 통해 쿼리 필드가 비어 있을 때 간격을 변경할 수 있는 옵션이 아닙니다. ([LOG-2917](#))

1.4.17.3. CVE

- [CVE-2022-1705](#)
- [CVE-2022-2526](#)
- [CVE-2022-29154](#)
- [CVE-2022-30631](#)
- [CVE-2022-32148](#)

- [CVE-2022-32206](#)
- [CVE-2022-32208](#)

1.4.18. Logging 5.5.0

이 릴리스에는 [OpenShift Logging](#) 버그 수정 릴리스 5.5.0 이 포함되어 있습니다.

1.4.18.1. 기능 개선

- 이번 업데이트를 통해 동일한 Pod 내의 다른 컨테이너에서 다른 인덱스로 구조화된 로그를 전달할 수 있습니다. 이 기능을 사용하려면 다중 컨테이너 지원을 사용하여 파이프라인을 구성하고 Pod에 주석을 달아야 합니다. ([LOG-1296](#))

중요

JSON 형식의 로그는 애플리케이션에 따라 다릅니다. 너무 많은 인덱스를 생성하면 성능에 영향을 미치기 때문에 이 기능을 사용하여 호환되지 않는 JSON 형식의 로그 인덱스를 생성할 수 있습니다. 쿼리를 사용하여 서로 다른 네임스페이스 또는 호환되는 JSON 형식의 애플리케이션을 분리합니다.
- 이번 업데이트를 통해 Kubernetes 공통 레이블인 `app.kubernetes.io/component`, `app.kubernetes.io/managed-by`, `app.kubernetes.io/part-of`, `app.kubernetes.io/version` 을 사용하여 Elasticsearch 출력으로 로그를 필터링할 수 있습니다. Elasticsearch가 아닌 출력 유형은 `kubernetes.labels` 에 포함된 모든 레이블을 사용할 수 있습니다. ([LOG-2388](#))
- 이번 업데이트를 통해 AWS STS(Security Token Service)가 활성화된 클러스터는 STS 인증을 사용하여 로그를 Amazon NetNamespace로 전달할 수 있습니다. ([LOG-1976](#))
- 이번 업데이트를 통해 Loki Operator 및 Vector 수집기가 기술 프리뷰에서 일반 가용성으로 이동합니다. 이전 릴리스와의 전체 기능 패리티가 보류 중이며 일부 API는 기술 프리뷰로 남아 있습니다. 자세한 내용은 [LokiStack](#)을 사용한 로깅 섹션을 참조하십시오.

1.4.18.2. 버그 수정

- 이번 업데이트 이전에는 로그를 **Amazon ONTAP**에 전달하도록 구성된 클러스터는 거부된 로그 파일을 임시 스토리지로 전달하여 시간이 지남에 따라 클러스터 불안정을 초래했습니다. 이번 업데이트를 통해 모든 스토리지 옵션의 청크 백업이 비활성화되어 문제를 해결합니다. ([LOG-2746](#))
- 이번 업데이트 이전에는 **Operator**가 더 이상 사용되지 않고 향후 **OpenShift Container Platform** 버전에서 제거될 계획인 일부 API 버전을 사용하고 있었습니다. 이번 업데이트에서는 종속 항목을 지원되는 API 버전으로 이동합니다. ([LOG-2656](#))
- 이번 업데이트 이전에는 다중 줄 오류 탐지를 위해 구성된 여러 **ClusterLogForwarder** 파이프라인으로 인해 수집기가 크래시 루프 복구 오류 상태로 전환되었습니다. 이번 업데이트에서는 여러 구성 섹션에 동일한 고유 ID가 있는 문제가 해결되었습니다. ([LOG-2241](#))
- 이번 업데이트 이전에는 수집기에서 **UTF-8** 이외의 기호를 **Elasticsearch** 스토리지 로그에 저장할 수 없었습니다. 이번 업데이트를 통해 수집기는 **UTF-8** 이외의 기호를 인코딩하여 문제를 해결합니다. ([LOG-2203](#))
- 이번 업데이트 이전에는 **Kibana**에 올바르지 않은 문자가 잘못 표시되었습니다. 이번 업데이트를 통해 **Kibana**가 모든 유효한 **UTF-8** 기호를 올바르게 표시합니다. ([LOG-2784](#))

1.4.18.3. CVE

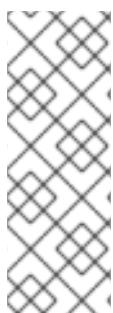
- [CVE-2021-38561](#)
- [CVE-2022-1012](#)
- [CVE-2022-1292](#)
- [CVE-2022-1586](#)
- [CVE-2022-1785](#)

- [CVE-2022-1897](#)
- [CVE-2022-1927](#)
- [CVE-2022-2068](#)
- [CVE-2022-2097](#)
- [CVE-2022-21698](#)
- [CVE-2022-30631](#)
- [CVE-2022-32250](#)

2장. 지원

이 문서에 설명된 구성 옵션만 로깅에 지원됩니다.

다른 구성 옵션은 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 **Operator**는 차이점을 조정하도록 설계되었으므로 변경 사항을 덮어씁니다.



참고

OpenShift Container Platform 설명서에 설명되어 있지 않은 구성을 수행해야 하는 경우 **Red Hat OpenShift Logging Operator**를 **Unmanaged** 상태로 설정해야 합니다. 관리되지 않는 로깅 인스턴스는 지원되지 않으며 **Managed**로 상태를 반환할 때까지 업데이트를 받지 않습니다.



참고

로깅은 핵심 **OpenShift Container Platform**과 별도의 릴리스 주기와 함께 설치 가능한 구성 요소로 제공됩니다. **Red Hat OpenShift Container Platform** 라이프 사이클 정책은 릴리스 호환성에 대해 간단히 설명합니다.

Red Hat OpenShift에 대한 로깅은 애플리케이션, 인프라 및 감사 로그의 의견이 지정된 수집기 및 노멀라이저입니다. 지원되는 다양한 시스템으로 로그를 전달하는 데 사용됩니다.

로깅은 다음과 같습니다.

- 대규모 로그 수집 시스템
- **SIEM(Security Information and Event Monitoring)** 준수
- 기록 또는 장기 로그 보존 또는 스토리지

- 보장된 로그 싱크
- 보안 스토리지 - 감사 로그는 기본적으로 저장되지 않습니다.

2.1. 지원되는 API 사용자 정의 리소스 정의

LokiStack 개발이 진행 중입니다. 현재 일부 API는 지원되지 않습니다.

표 2.1. Loki API 지원 상태

CRD(CustomResourceDefinition)	ApiVersion	지원 상태
LokiStack	lokistack.loki.grafana.com/v1	5.5에서 지원됨
RulerConfig	rulerconfig.loki.grafana/v1	5.7에서 지원
AlertingRule	alertingrule.loki.grafana/v1	5.7에서 지원
RecordingRule	recordingrule.loki.grafana/v1	5.7에서 지원

2.2. 지원되지 않는 로깅 구성

다음 구성 요소를 수정 하려면 **Red Hat OpenShift Logging Operator**를 **Unmanaged** 상태로 설정해야 합니다.

- **Elasticsearch CR**(사용자 정의 리소스)
- **Kibana 배포**
- **fluent.conf** 파일
- **Fluentd 데몬 세트**

Elasticsearch 배포 파일을 수정하려면 **OpenShift Elasticsearch Operator**를 **Unmanaged** 상태로 설정해야 합니다.

명시적으로 지원되지 않는 경우는 다음과 같습니다.

- 기본 로그 회전 구성. 기본 로그 회전 구성을 수정할 수 없습니다.
- 수집된 로그 위치 구성. 로그 수집기 출력 파일의 위치는 기본적으로 `/var/log/fluentd/fluentd.log`입니다.
- 제한 로그 수집. 로그 수집기에서 로그를 읽는 속도를 조절할 수 없습니다.
- 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.

2.3. 관리되지 않는 **OPERATOR**에 대한 지원 정책

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

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

다음과 같은 방법으로 **Operator**를 **Unmanaged** 상태로 설정할 수 있습니다.

- 개별 **Operator** 구성

개별 **Operator**는 구성에 **managementState** 매개변수가 있습니다. **Operator**에 따라 다양한

방식으로 이 매개변수에 액세스할 수 있습니다. 예를 들어, **Red Hat OpenShift Logging Operator**는 관리 대상인 사용자 정의 리소스(CR)를 수정하여 이를 수행하는 반면 **Cluster Samples Operator**는 클러스터 전체의 구성 리소스를 사용합니다.

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



주의

개별 Operator를 **Unmanaged** 상태로 변경하면 특정 구성 요소 및 기능이 지원되지 않습니다. 지원을 계속하려면 보고된 문제를 **Managed** 상태에서 재현해야 합니다.

-

Cluster Version Operator(CVO) 재정의

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

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



주의

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

2.4. RED HAT 지원을 위한 로깅 테이터 수집

지원 케이스를 열 때 클러스터에 대한 디버깅 정보를 Red Hat 지원에 제공하는 것이 좋습니다.

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

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



참고

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

2.4.1. must-gather 툴 정보

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

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

2.4.2. 로깅 데이터 수집

oc adm must-gather CLI 명령을 사용하여 로깅에 대한 정보를 수집할 수 있습니다.

프로세스

must-gather로 로깅 정보를 수집하려면 다음을 수행합니다.

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

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

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

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

```
$ tar -cvaf must-gather.tar.gz must-gather.local.4157245944708210408
```
4. [Red Hat Customer Portal](#)에서 해당 지원 사례에 압축 파일을 첨부합니다.

3장. 로깅 문제 해결

3.1. 로깅 상태 보기

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

3.1.1. Red Hat OpenShift Logging Operator의 상태 보기

Red Hat OpenShift Logging Operator의 상태를 볼 수 있습니다.

사전 요구 사항

- Red Hat OpenShift Logging Operator 및 OpenShift Elasticsearch Operator가 설치되어 있습니다.

절차

- 다음 명령을 실행하여 **openshift-logging** 프로젝트로 변경합니다.

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

- 다음 명령을 실행하여 **ClusterLogging** 인스턴스 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:
        collector-2rhqp: ip-10-0-169-13.ec2.internal
        collector-6fgjh: ip-10-0-165-244.ec2.internal
```

```

    collector-6l2ff: ip-10-0-128-218.ec2.internal
    collector-54nx5: ip-10-0-139-30.ec2.internal
    collector-flpnn: ip-10-0-147-228.ec2.internal
    collector-n2frh: ip-10-0-157-45.ec2.internal
  pods:
    failed: []
    notReady: []
    ready:
      - collector-2rhqp
      - collector-54nx5
      - collector-6fgjh
      - collector-6l2ff
      - collector-flpnn
      - collector-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 Pod**에 대한 정보입니다.

4

Kibana Pod에 대한 정보.

3.1.1.1. 상태 메시지 예

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

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

출력 예

```
nodes:
- conditions:
  - lastTransitionTime: 2019-03-15T15:57:22Z
    message: Disk storage usage for node is 27.5gb (36.74%). Shards will be not
              be allocated on this node.
    reason: Disk Watermark Low
    status: "True"
```

```

type: NodeStorage
deploymentName: example-elasticsearch-clientdatamaster-0-1
upgradeStatus: {}

```

다음과 유사한 상태 메시지는 노드가 구성된 높은 워터마크를 초과했으며 **shard**가 다른 노드로 재배치됨을 나타냅니다.

출력 예

```

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

```

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

출력 예

```

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

```

```

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

```

다음과 유사한 상태 메시지는 요청한 PVC가 PV에 바인딩 할 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:
```

3.1.2. 로깅 구성 요소의 상태 보기

여러 로깅 구성 요소의 상태를 볼 수 있습니다.

사전 요구 사항

- Red Hat OpenShift Logging Operator 및 OpenShift Elasticsearch Operator가 설치되어 있습니다.

프로세스

- openshift-logging 프로젝트로 변경합니다.

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

- 로깅 환경의 상태 보기:

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

출력 예

```
Name: cluster-logging-operator
...
Conditions:
  Type     Status  Reason
  ----  -----  -----
  Available  True   MinimumReplicasAvailable
  Progressing True   NewReplicaSetAvailable
...
Events:
  Type    Reason          Age From            Message
  ----  -----  ----  -----
  Normal  ScalingReplicaSet 62m deployment-controller Scaled up replica set
cluster-logging-operator-574b8987df to 1----
```

3.

로깅 복제본 세트의 상태를 확인합니다.

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-3-5b6fdd8cfdf 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----
```

3.2. 로그 전달 문제 해결

3.2.1. Fluentd Pod 재배포

ClusterLogForwarder CR(사용자 정의 리소스)을 생성할 때 **Red Hat OpenShift Logging Operator** 가 **Fluentd Pod**를 자동으로 재배포하지 않으면 **Fluentd Pod**를 삭제하여 강제로 재배포할 수 있습니다.

사전 요구 사항

-

ClusterLogForwarder CR(사용자 정의 리소스) 오브젝트가 생성되어 있습니다.

프로세스

- 다음 명령을 실행하여 Fluentd Pod를 삭제하여 강제로 재배포합니다.

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

3.2.2. Loki 속도 제한 오류 문제 해결

로그 전달자 API에서 속도 제한을 초과하는 대규모 메시지 블록을 Loki로 전달하면 Loki는 속도 제한(429) 오류를 생성합니다.

이러한 오류는 정상적인 작동 중에 발생할 수 있습니다. 예를 들어 이미 일부 로그가 있는 클러스터에 로깅을 추가할 때 로깅이 기존 로그 항목을 모두 수집하는 동안 속도 제한 오류가 발생할 수 있습니다. 이 경우 새 로그 추가 속도가 총 속도 제한보다 작으면 기록 데이터가 결국 수집되고 사용자 개입 없이도 속도 제한 오류가 해결됩니다.

속도 제한 오류가 계속 발생하는 경우 **LokiStack CR**(사용자 정의 리소스)을 수정하여 문제를 해결할 수 있습니다.



중요

LokiStack CR은 Grafana 호스팅 Loki에서 사용할 수 없습니다. 이는 Grafana 호스팅 Loki 서버에는 적용되지 않습니다.

조건

- Log Forwarder API는 로그를 Loki로 전달하도록 구성되어 있습니다.
 - 시스템에서 2MB보다 큰 메시지 블록을 Loki로 보냅니다. 예를 들면 다음과 같습니다.

```
"values": [{"id": "1630410392689800468", "type": "Event", "apiVersion": "1.7.4", "received_at": "2021-08-31T11:46:32.800Z", "version": "1.7.4", "timestramp": "2021-08-31T11:46:32.800Z"}]
```

```
31T11:46:32.799692+00:00\","viaq_index_name\":\"audit-
write\" ,\"viaq_msg_id\":\"MzFjYjJkZjlNjY0MC00YWU4LWIwMTEtNGNmM2E5ZmViMGU
4\" ,\"log_type\":\"audit\" }"]}]}
```

- `oc logs -n openshift-logging -l component=collector` 를 입력하면 클러스터의 수집기 로그에 다음 오류 메시지 중 하나가 포함된 행이 표시됩니다.

429 Too Many Requests Ingestion rate limit exceeded

Vector 오류 메시지의 예

```
2023-08-25T16:08:49.301780Z WARN sink{component_kind="sink"
component_id=default_loki_infra component_type=loki
component_name=default_loki_infra}: vector::sinks::util::retries: Retrying after error.
error=Server responded with an error: 429 Too Many Requests
internal_log_rate_limit=true
```

Fluentd 오류 메시지의 예

```
2023-08-30 14:52:15 +0000 [warn]: [default_loki_infra] failed to flush the buffer.
retry_times=2 next_retry_time=2023-08-30 14:52:19 +0000
chunk="604251225bf5378ed1567231a1c03b8b"
error_class=Fluent::Plugin::LokiOutput::LogPostError error="429 Too Many Requests
Ingestion rate limit exceeded for user infrastructure (limit: 4194304 bytes/sec) while
attempting to ingest '4082' lines totaling '7820025' bytes, reduce log volume or contact
your Loki administrator to see if the limit can be increased\n"
```

이 오류는 수신 끝점에도 표시됩니다. 예를 들어 **LokiStack ingester Pod**에서 다음을 수행합니다.

Loki ingester 오류 메시지의 예

```
level=warn ts=2023-08-30T14:57:34.155592243Z caller=grpc_logging.go:43
```

```
duration=1.434942ms method=/logproto.Pusher/Push err="rpc error: code = Code(429)
desc = entry with timestamp 2023-08-30 14:57:32.012778399 +0000 UTC ignored,
reason: 'Per stream rate limit exceeded (limit: 3MB/sec) while attempting to ingest for
stream'
```

프로세스

- LokiStack CR에서 **ingestionBurstSize** 및 **ingestionRate** 필드를 업데이트합니다.

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
  limits:
    global:
      ingestion:
        ingestionBurstSize: 16 ①
        ingestionRate: 8 ②
# ...
```

①

ingestionBurstSize 필드는 배포자 복제본당 최대 로컬 속도 제한 샘플 크기를 **MB**로 정의합니다. 이 값은 하드 제한입니다. 이 값을 단일 푸시 요청에 예상되는 최대 로그 크기로 설정합니다. **ingestionBurstSize** 값보다 큰 단일 요청은 허용되지 않습니다.

②

ingestionRate 필드는 초당 수집된 샘플의 최대 양(**MB**)에 대한 소프트 제한입니다. 로그 비율이 제한을 초과하는 경우 속도 제한 오류가 발생하지만 수집기는 로그를 다시 시도합니다. 총 평균이 제한보다 작으면 사용자 개입 없이 시스템을 복구하고 오류가 해결됩니다.

3.3. 로깅 경고 문제 해결

다음 절차를 사용하여 클러스터의 로깅 경고 문제를 해결할 수 있습니다.

3.3.1. Elasticsearch 클러스터 상태가 빨간색

하나 이상의 기본 **shard**와 해당 복제본이 노드에 할당되지 않습니다. 다음 절차에 따라 이 경고 문제를

해결합니다.

작은 정보

이 문서의 일부 명령은 **\$ES_POD_NAME** 쉘 변수를 사용하여 **Elasticsearch Pod**를 참조합니다. 이 문서에서 직접 명령을 복사하여 붙여넣려면 이 변수를 **Elasticsearch** 클러스터에 유효한 값으로 설정해야 합니다.

다음 명령을 실행하여 사용 가능한 **Elasticsearch Pod**를 나열할 수 있습니다.

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

나열된 **Pod** 중 하나를 선택하고 다음 명령을 실행하여 **\$ES_POD_NAME** 변수를 설정합니다.

```
$ export ES_POD_NAME=<elasticsearch_pod_name>
```

이제 명령에 **\$ES_POD_NAME** 변수를 사용할 수 있습니다.

프로세스

1.

Elasticsearch 클러스터 상태를 확인하고 다음 명령을 실행하여 클러스터 상태가 빨간색인지 확인합니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES_POD_NAME -- health
```

2.

다음 명령을 실행하여 클러스터에 참여한 노드를 나열합니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES_POD_NAME \
-- es_util --query=_cat/nodes?v
```

3.

다음 명령을 실행하여 **Elasticsearch Pod**를 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.

다음 명령을 실행하고 출력을 관찰하여 Elasticsearch에 선택한 마스터 노드가 있는지 확인합니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES_POD_NAME \
-- es_util --query=_cat/master?v
```

b.

다음 명령을 실행하고 출력을 관찰하여 선택한 마스터 노드의 Pod 로그를 검토합니다.

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

c.

다음 명령을 실행하고 출력을 관찰하여 클러스터에 참여하지 않은 노드의 로그를 확인합니다.

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

5.

모든 노드가 클러스터에 참여한 경우 다음 명령을 실행하고 출력을 관찰하여 클러스터가 복구 프로세스 중인지 확인합니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES_POD_NAME \
-- es_util --query=_cat/recovery?active_only=true
```

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

6.

다음 명령을 실행하고 출력을 관찰하여 보류 중인 작업이 있는지 확인합니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES_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 보이면 다음 명령을 실행하고 출력을 관찰하여

`cluster.routing.allocation.enable` 값이 `none` 으로 설정되어 있는지 확인합니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES_POD_NAME \
-- es_util --query=_cluster/settings?pretty
```

9.

`cluster.routing.allocation.enable` 값이 `none` 으로 설정된 경우 다음 명령을 실행하여 `all` 로 설정합니다.

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

10.

다음 명령을 실행하고 출력을 관찰하여 인덱스가 빨간색인지 확인합니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES_POD_NAME \
-- es_util --query=_cat/indices?v
```

11.

인덱스가 빨간색이면 다음 단계를 수행하여 지웁니다.

a.

다음 명령을 실행하여 캐시를 지웁니다.

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

b.

다음 명령을 실행하여 최대 할당 제시도 횟수를 늘립니다.

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

c.

다음 명령을 실행하여 스크롤 항목을 모두 삭제합니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES_POD_NAME \
-- es_util --query=_search	scroll/_all -X DELETE
```

d.

다음 명령을 실행하여 시간 초과를 늘립니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES_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.

다음 명령을 실행하여 빨간색 인덱스 이름을 확인합니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES_POD_NAME \
-- es_util --query=_cat/indices?v
```

b.

다음 명령을 실행하여 빨간색 인덱스를 삭제합니다.

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

13.

빨간색 인덱스가 없고 클러스터 상태가 빨간색이면 데이터 노드에서 지속적으로 처리 로드가 높은지 확인합니다.

a.

다음 명령을 실행하여 Elasticsearch JVM 힙 사용량이 높은지 확인합니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES_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. CPU 활용에 대한 자세한 내용은 [OpenShift Container Platform "모듈 대시보드 검토" 설명서](#)를 참조하십시오.

추가 리소스



[모니터링 대시보드 검토](#)



[빨간색 또는 노란색 클러스터 상태 설정](#)

3.3.2. Elasticsearch 클러스터 상태가 노란색임

하나 이상의 기본 shard의 복제본 shard는 노드에 할당되지 않습니다. ClusterLogging 사용자 정의 리소스(CR)에서 nodeCount 값을 조정하여 노드 수를 늘립니다.

추가 리소스

-

[빨간색 또는 노란색 클러스터 상태 설정](#)

3.3.3. Elasticsearch 노드 디스크 낮은 워터마크 도달

Elasticsearch는 낮은 워터마크에 도달하는 노드에 shard를 할당하지 않습니다.

작은 정보

이 문서의 일부 명령은 \$ES_POD_NAME 쉘 변수를 사용하여 Elasticsearch Pod를 참조합니다. 이 문서에서 직접 명령을 복사하여 붙여넣려면 이 변수를 Elasticsearch 클러스터에 유효한 값으로 설정해야 합니다.

다음 명령을 실행하여 사용 가능한 Elasticsearch Pod를 나열할 수 있습니다.

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

나열된 Pod 중 하나를 선택하고 다음 명령을 실행하여 \$ES_POD_NAME 변수를 설정합니다.

```
$ export ES_POD_NAME=<elasticsearch_pod_name>
```

이제 명령에 \$ES_POD_NAME 변수를 사용할 수 있습니다.

프로세스

1.

다음 명령을 실행하여 Elasticsearch가 배포된 노드를 식별합니다.

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

2.

다음 명령을 실행하여 할당되지 않은 shard가 있는지 확인합니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES_POD_NAME \
-- es_util --query=_cluster/health?pretty | grep unassigned_shards
```

3.

할당되지 않은 shard가 있는 경우 다음 명령을 실행하여 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.

명령 출력에서 Use 열을 확인하여 해당 노드에서 사용된 디스크 백분율을 확인합니다.

출력 예

```
elasticsearch-cdm-kcrsda6l-1-586cc95d4f-h8zq8
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme1n1    19G  522M  19G  3% /elasticsearch/persistent
elasticsearch-cdm-kcrsda6l-2-5b548fc7b-cwwk7
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme2n1    19G  522M  19G  3% /elasticsearch/persistent
elasticsearch-cdm-kcrsda6l-3-5dfc884d99-59tjw
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme3n1    19G  528M  19G  3% /elasticsearch/persistent
```

사용된 디스크 백분율이 85%를 초과하는 경우 노드가 낮은 워터마크를 초과하여 더 이상 이 노드에 shard를 할당할 수 없습니다.

5.

현재 redundancyPolicy 를 확인하려면 다음 명령을 실행합니다.

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

클러스터에서 **ClusterLogging** 리소스를 사용하는 경우 다음 명령을 실행합니다.

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

클러스터 **redundancyPolicy** 값이 **SingleRedundancy** 값보다 크면 **SingleRedundancy** 값으로 설정하고 이 변경 사항을 저장합니다.

6.

이전 단계에서 문제가 해결되지 않으면 이전 인덱스를 삭제합니다.

a.

다음 명령을 실행하여 **Elasticsearch**의 모든 인덱스의 상태를 확인합니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES_POD_NAME -- indices
```

b.

삭제할 수 있는 이전 인덱스를 확인합니다.

c.

다음 명령을 실행하여 인덱스를 삭제합니다.

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

3.3.4. 높은 워터마크에 도달한 **Elasticsearch** 노드 디스크

Elasticsearch는 워터마크 임계값 제한을 초과하지 않은 디스크 사용량이 낮은 노드로 높은 워터마크에 도달한 노드에서 **shard**를 재배치하려고 합니다.

특정 노드에 **shard**를 할당하려면 해당 노드에서 일부 공간을 확보해야 합니다. 디스크 공간을 늘릴 수 없는 경우 클러스터에 새 데이터 노드를 추가하거나 총 클러스터 중복 정책을 줄입니다.

작은 정보

이 문서의 일부 명령은 **\$ES_POD_NAME** 쉘 변수를 사용하여 Elasticsearch Pod를 참조합니다. 이 문서에서 직접 명령을 복사하여 붙여넣려면 이 변수를 Elasticsearch 클러스터에 유효한 값으로 설정해야 합니다.

다음 명령을 실행하여 사용 가능한 Elasticsearch Pod를 나열할 수 있습니다.

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

나열된 Pod 중 하나를 선택하고 다음 명령을 실행하여 **\$ES_POD_NAME** 변수를 설정합니다.

```
$ export ES_POD_NAME=<elasticsearch_pod_name>
```

이제 명령에 **\$ES_POD_NAME** 변수를 사용할 수 있습니다.

프로세스

1.

다음 명령을 실행하여 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 $ES_POD_NAME \
-- es_util --query=_cluster/health?pretty | grep relocating_shards
```

명령 출력에 shard 재배치가 표시되면 높은 워터마크가 초과되었습니다. 높은 워터마크의 기본값은 **90%**입니다.

4.

모든 노드의 디스크 공간을 늘립니다. 디스크 공간을 늘릴 수 없는 경우 클러스터에 새 데이터 노드를 추가하거나 총 클러스터 중복 정책을 줄입니다.

5.

현재 **redundancyPolicy** 를 확인하려면 다음 명령을 실행합니다.

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

클러스터에서 **ClusterLogging** 리소스를 사용하는 경우 다음 명령을 실행합니다.

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

클러스터 **redundancyPolicy** 값이 **SingleRedundancy** 값보다 크면 **SingleRedundancy** 값으로 설정하고 이 변경 사항을 저장합니다.

6.

이전 단계에서 문제가 해결되지 않으면 이전 인덱스를 삭제합니다.

a.

다음 명령을 실행하여 **Elasticsearch**의 모든 인덱스의 상태를 확인합니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES_POD_NAME -- indices
```

b.

삭제할 수 있는 이전 인덱스를 확인합니다.

c.

다음 명령을 실행하여 인덱스를 삭제합니다.

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

3.3.5. Elasticsearch 노드 디스크 플리드 워터마크에 도달했습니다

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.

다음 절차에 따라 이 경고 문제를 해결합니다.

작은 정보

이 문서의 일부 명령은 `$ES_POD_NAME` 쉘 변수를 사용하여 Elasticsearch Pod를 참조합니다. 이 문서에서 직접 명령을 복사하여 붙여넣려면 이 변수를 Elasticsearch 클러스터에 유효한 값으로 설정해야 합니다.

다음 명령을 실행하여 사용 가능한 Elasticsearch Pod를 나열할 수 있습니다.

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

나열된 Pod 중 하나를 선택하고 다음 명령을 실행하여 `$ES_POD_NAME` 변수를 설정합니다.

```
$ export ES_POD_NAME=<elasticsearch_pod_name>
```

이제 명령에 `$ES_POD_NAME` 변수를 사용할 수 있습니다.

프로세스

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 **Avail** 열을 확인하여 해당 노드에서 사용 가능한 디스크 공간을 확인합니다.

출력 예

```
elasticsearch-cdm-kcrsda6l-1-586cc95d4f-h8zq8
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme1n1    19G  522M  19G  3% /elasticsearch/persistent
elasticsearch-cdm-kcrsda6l-2-5b548fc7b-cwwk7
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme2n1    19G  522M  19G  3% /elasticsearch/persistent
elasticsearch-cdm-kcrsda6l-3-5dfc884d99-59tjw
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme3n1    19G  528M  19G  3% /elasticsearch/persistent
```

3.

모든 노드의 디스크 공간을 늘립니다. 디스크 공간을 늘릴 수 없는 경우 클러스터에 새 데이터 노드를 추가하거나 총 클러스터 중복 정책을 줄입니다.

4.

현재 **redundancyPolicy** 를 확인 하려면 다음 명령을 실행합니다.

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

클러스터에서 **ClusterLogging** 리소스를 사용하는 경우 다음 명령을 실행합니다.

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

클러스터 **redundancyPolicy** 값이 **SingleRedundancy** 값보다 크면 **SingleRedundancy** 값으로 설정하고 이 변경 사항을 저장합니다.

5.

이전 단계에서 문제가 해결되지 않으면 이전 인덱스를 삭제합니다.

a.

다음 명령을 실행하여 **Elasticsearch**의 모든 인덱스의 상태를 확인합니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES POD_NAME -- indices
```

b.

삭제할 수 있는 이전 인덱스를 확인합니다.

c.

다음 명령을 실행하여 인덱스를 삭제합니다.

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

6.

디스크 공간을 계속 확보하고 모니터링합니다. 사용된 디스크 공간이 90% 미만으로 떨어지면 다음 명령을 실행하여 이 노드에 쓰기 차단을 해제합니다.

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

3.3.6. Elasticsearch JVM 힙 사용량이 높음

사용된 Elasticsearch 노드 JVM(Java 가상 머신) 힙 메모리는 75% 이상입니다. 힙 크기를 늘리는 것 이 좋습니다.

3.3.7. 집계된 로깅 시스템 CPU가 높음

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

3.3.8. Elasticsearch 프로세스 CPU가 높음

노드의 Elasticsearch 프로세스 CPU 사용량이 높습니다. 클러스터 노드의 CPU를 확인합니다. 더 많은 CPU 리소스를 노드에 할당하는 것이 좋습니다.

3.3.9. Elasticsearch 디스크 공간이 부족합니다.

Elasticsearch는 현재 디스크 사용량에 따라 향후 6시간 이내에 디스크 공간이 부족해질 것으로 예상됩니다. 다음 절차에 따라 이 경고 문제를 해결합니다.

프로세스

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 **Avail** 열을 확인하여 해당 노드에서 사용 가능한 디스크 공간을 확인합니다.

출력 예

```
elasticsearch-cdm-kcrsda6l-1-586cc95d4f-h8zq8
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme1n1    19G   522M  19G   3% /elasticsearch/persistent
elasticsearch-cdm-kcrsda6l-2-5b548fc7b-cwwk7
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme2n1    19G   522M  19G   3% /elasticsearch/persistent
elasticsearch-cdm-kcrsda6l-3-5dfc884d99-59tjw
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme3n1    19G   528M  19G   3% /elasticsearch/persistent
```

3.

모든 노드의 디스크 공간을 늘립니다. 디스크 공간을 늘릴 수 없는 경우 클러스터에 새 데이터 노드를 추가하거나 총 클러스터 중복 정책을 줄입니다.

4.

현재 **redundancyPolicy** 를 확인하려면 다음 명령을 실행합니다.

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

클러스터에서 **ClusterLogging** 리소스를 사용하는 경우 다음 명령을 실행합니다.

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

클러스터 **redundancyPolicy** 값이 **SingleRedundancy** 값보다 크면 **SingleRedundancy** 값으로 설정하고 이 변경 사항을 저장합니다.

5.

이전 단계에서 문제가 해결되지 않으면 이전 인덱스를 삭제합니다.

a.

다음 명령을 실행하여 Elasticsearch의 모든 인덱스의 상태를 확인합니다.

```
$ oc exec -n openshift-logging -c elasticsearch $ES_POD_NAME -- indices
```

b.

삭제할 수 있는 이전 인덱스를 확인합니다.

c.

다음 명령을 실행하여 인덱스를 삭제합니다.

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

추가 리소스

-

[빨간색 또는 노란색 클러스터 상태 설정](#)

3.3.10. Elasticsearch FileDescriptor 사용량이 높음

현재 사용 추세를 기준으로 노드의 예상 파일 설명자 수가 충분하지 않습니다. [Elasticsearch File Descriptors](#) 문서에 설명된 대로 각 노드의 `max_file_descriptors` 값을 확인합니다.

3.4. ELASTICSEARCH 로그 저장소의 상태 보기

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

3.4.1. Elasticsearch 로그 저장소의 상태 보기

Elasticsearch 로그 저장소의 상태를 볼 수 있습니다.

사전 요구 사항

-

[Red Hat OpenShift Logging Operator](#) 및 [OpenShift Elasticsearch Operator](#)가 설치되어 있습니다.

절차

1.

다음 명령을 실행하여 **openshift-logging** 프로젝트로 변경합니다.

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

2.

상태를 보려면 다음을 수행합니다.

a.

다음 명령을 실행하여 **Elasticsearch** 로그 저장소 인스턴스의 이름을 가져옵니다.

```
$ oc get Elasticsearch
```

출력 예

NAME	AGE
elasticsearch	5h9m

b.

다음 명령을 실행하여 **Elasticsearch** 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 스템자에 나타납니다.

②

Elasticsearch 로그 저장소의 상태:

- 활성 기본 **shard** 수입니다.
- 활성 **shard** 수입니다.
- 초기화 중인 **shard** 수입니다.
- **Elasticsearch** 로그 저장소 데이터 노드의 수입니다.
- 총 **Elasticsearch** 로그 저장소 노드 수입니다.
- 보류 중인 작업 수입니다.
- **Elasticsearch** 로그 저장소 상태: **green,red,yellow**.
- 할당되지 않은 **shard** 수

3

존재하는 경우 모든 상태 조건. **Elasticsearch** 로그 저장소 상태는 Pod를 배치할 수 없는 경우 스케줄러의 이유를 나타냅니다. 다음 조건과 관련된 모든 이벤트가 표시됩니다.

- 컨테이너는 **Elasticsearch** 로그 저장소 및 프록시 컨테이너를 모두 대기합니다.
- 컨테이너는 **Elasticsearch** 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 **Elasticsearch** 로그 저장소 노드

5

실패한, **notReady** 또는 **ready** 상태에 나열된 클러스터의 **Elasticsearch** 로그 저장소 클라이언트, 데이터 및 마스터 Pod

3.4.1.1. 상태 메시지 예

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

다음 상태 메시지는 노드가 구성된 낮은 워터마크를 초과했으며 이 노드에 **shard**가 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 **shard**가 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 CR(사용자 정의 리소스)의 Elasticsearch 로그 저장소 노드 선택기가 클러스터의 노드와 일치하지 않음을 나타냅니다.

```

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

```

다음 상태 메시지는 Elasticsearch 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: Unschedulable
    status: True
    type: Unschedulable

```

다음 상태 메시지는 Elasticsearch 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 플레이 노드가 너무 많음을 나타냅니다.

```

status:
clusterHealth: green

```

conditions:

- lastTransitionTime: '2019-04-17T20:12:34Z'
- message: >-

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

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: StorageStructureChangegnored
 - status: 'True'
 - type: StorageStructureChangegnored

reason 및 **type** 필드는 지원되지 않는 변경 유형을 지정합니다.

StorageClassNameChangegnored

스토리지 클래스 이름에 대한 지원되지 않는 변경 사항입니다.

StorageSizeChangegnored

스토리지 크기에 대한 지원되지 않는 변경 사항입니다.

StorageStructureChangegnored

임시 스토리지 구조와 영구저장장치 구조 간에는 지원되지 않는 변경 사항입니다.



중요

임시 스토리지에서 영구 스토리지로 전환하도록 ClusterLogging CR을 구성하려고 하면 OpenShift Elasticsearch Operator는 PVC(영구 볼륨 클레임)를 생성하지만 PV(영구 볼륨)를 생성하지 않습니다. StorageStructureChangegnored 상태를 지우려면 ClusterLogging CR로 변경 사항을 취소하고 PVC를 삭제해야 합니다.

3.4.2. 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
S4QANnf1QP6NgCegfnrbQ 3 1 119926 0 157 78
green open audit-000001
8_EQx77iQCSTzFOXtxRqFw 3 1 0 0 0 0
green open .security iDjscH7aSUGhIdq0LheLBQ
1 1 5 0 0 0
green open .kibana_-377444158_kubeadmin
yBywZ9GfSrKebz5gWBZbjw 3 1 1 0 0 0
green open infra-000001
```

```

z6Dpe__ORgiopEpW6YI44A 3 1 871000 0 874 436
green open app-000001 hlrazQCeSISewG3c2VlvsQ
3 1 2453 0 3 1
green open .kibana_1 JCitcBMSQxKOvlq6iQW6wg
1 1 0 0 0 0
green open .kibana_-1595131456_user1 gIYFIEGRRe-
ka0W3okS-mQ 3 1 1 0 0 0

```

로그 저장소 Pod

로그 저장소를 호스팅하는 **Pod**의 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-  

rhe18@sha256:4265742c7cdd85359140e2d7d703e4311b6497eec7676957f455d6908e7  

b1c25  

    Readiness: exec [/usr/share/elasticsearch/probe/readiness.sh] delay=10s  

timeout=30s period=5s #success=1 #failure=3  

....  

Events: <none>
```

3.4.3. Elasticsearch 클러스터 상태

OpenShift Container Platform 웹 콘솔의 **Observe** 섹션에 있는 대시보드에는 **Elasticsearch** 클러스터의 상태가 표시됩니다.

OpenShift Elasticsearch 클러스터의 상태를 가져오려면 `<cluster_url>/monitoring/dashboards/grafana-dashboard-cluster-logging`에 있는 OpenShift Container Platform 웹 콘솔의 **Observe** 섹션에서 대시보드를 참조하십시오.

Elasticsearch 상태 필드

eo_elasticsearch_cr_cluster_management_state

Elasticsearch 클러스터가 관리 상태인지 또는 관리되지 않는 상태에 있는지를 표시합니다. 예를 들면 다음과 같습니다.

```
eo_elasticsearch_cr_cluster_management_state{state="managed"} 1
eo_elasticsearch_cr_cluster_management_state{state="unmanaged"} 0
```

eo_elasticsearch_cr_restart_total

인증서 재시작, 롤링 재시작 또는 예약된 재시작을 위해 Elasticsearch 노드가 다시 시작된 횟수를 표시합니다. 예를 들면 다음과 같습니다.

```
eo_elasticsearch_cr_restart_total{reason="cert_restart"} 1
eo_elasticsearch_cr_restart_total{reason="rolling_restart"} 1
eo_elasticsearch_cr_restart_total{reason="scheduled_restart"} 3
```

es_index_namespaces_total

Elasticsearch 인덱스 네임스페이스의 총 수를 표시합니다. 예를 들면 다음과 같습니다.

```
Total number of Namespaces.
es_index_namespaces_total 5
```

es_index_document_count

각 네임스페이스에 대한 레코드 수가 표시됩니다. 예를 들면 다음과 같습니다.

```
es_index_document_count{namespace="namespace_1"} 25
es_index_document_count{namespace="namespace_2"} 10
es_index_document_count{namespace="namespace_3"} 5
```

"Secret Elasticsearch 필드가 누락되었거나 비어 있음" 메시지

Elasticsearch에 admin-cert,admin-key,logging-es.crt 또는 logging-es.key 파일이 없는 경우 대시보드에는 다음 예와 유사한 상태 메시지가 표시됩니다.

```
message": "Secret \"elasticsearch\" fields are either missing or empty: [admin-cert, admin-key, logging-es.crt, logging-es.key]",
"reason": "Missing Required Secrets",
```

4장. 로깅 정보

클러스터 관리자는 **OpenShift Container Platform** 클러스터에 로깅을 배포하고 이를 사용하여 노드 시스템 감사 로그, 애플리케이션 컨테이너 로그 및 인프라 로그를 수집하고 집계할 수 있습니다. 온-클러스터, **Red Hat** 관리 로그 스토리지를 포함하여 선택한 로그 출력에 로그를 전달할 수 있습니다. 배포된 로그 스토리지 솔루션에 따라 **OpenShift Container Platform** 웹 콘솔 또는 **Kibana** 웹 콘솔에서 로그 데이터를 시각화할 수도 있습니다.



참고

Kibana 웹 콘솔은 향후 로깅 릴리스에서 더 이상 사용되지 않습니다.

OpenShift Container Platform 클러스터 관리자는 **Operator**를 사용하여 로깅을 배포할 수 있습니다. 자세한 내용은 [로깅 설치](#)를 참조하십시오.

Operator는 로깅 배포, 업그레이드 및 유지보수를 담당합니다. **Operator**가 설치되면 **ClusterLogging** 사용자 정의 리소스(**CR**)를 생성하여 로깅 **Pod** 및 로깅을 지원하는 데 필요한 기타 리소스를 예약할 수 있습니다. **ClusterLogForwarder CR**을 생성하여 수집되는 로그, 변환 방법, 전달되는 위치를 지정할 수도 있습니다.



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 **Kibana**에서 감사 로그를 보려면 감사 로그를 기본 내부 **Elasticsearch** 로그 저장소로 보내려면 로그 저장소에 감사 로그 전달에 설명된 대로 [로그 전달 API](#)를 사용해야 합니다.

4.1. 로깅 아키텍처

로깅의 주요 구성 요소는 다음과 같습니다.

수집기

수집기는 데몬 세트로 각 **OpenShift Container Platform** 노드에 **Pod**를 배포합니다. 각 노드에서 로그 데이터를 수집하고 데이터를 변환한 다음 구성된 출력으로 전달합니다. **Vector** 수집기 또는 레거시 **Fluentd** 수집기를 사용할 수 있습니다.



참고

Fluentd는 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다. **Red Hat**은 현재 릴리스 라이프사이클 동안 이 기능에 대한 버그 수정 및 지원을 제공하지만 이 기능은 더 이상 개선 사항을 받지 않습니다. **Fluentd** 대신 **Vector**를 사용할 수 있습니다.

로그 저장소

로그 저장소는 분석을 위해 로그 데이터를 저장하고 로그 전달자의 기본 출력입니다. 기본 **LokiStack** 로그 저장소, 레거시 **Elasticsearch** 로그 저장소를 사용하거나 로그를 추가 외부 로그 저장소로 전달할 수 있습니다.



참고

로깅 5.9 릴리스에는 업데이트된 **OpenShift Elasticsearch Operator** 버전이 포함되어 있지 않습니다. 현재 **Logging 5.8**과 함께 릴리스된 **OpenShift Elasticsearch Operator**를 사용하는 경우 로깅 5.8의 EOL까지 로깅에서 계속 작동합니다. **OpenShift Elasticsearch Operator**를 사용하여 기본 로그 스토리지를 관리하는 대신 **Loki Operator**를 사용할 수 있습니다. 로깅 라이프사이클 날짜에 대한 자세한 내용은 [Platform Agnostic Operators](#)를 참조하십시오.

시각화

UI 구성 요소를 사용하여 로그 데이터의 시각적 표현을 볼 수 있습니다. UI는 저장된 로그를 검색, 쿼리 및 볼 수 있는 그래픽 인터페이스를 제공합니다. **OpenShift Container Platform** 웹 콘솔 UI는 **OpenShift Container Platform** 콘솔 플러그인을 활성화하면 제공됩니다.



참고

Kibana 웹 콘솔은 향후 로깅 릴리스에서 더 이상 사용되지 않습니다.

로깅은 컨테이너 로그 및 노드 로그를 수집합니다. 이는 다음과 같은 유형으로 분류됩니다.

애플리케이션 로그

인프라 컨테이너 애플리케이션을 제외하고 클러스터에서 실행 중인 사용자 애플리케이션에 의해 생성된 컨테이너 로그입니다.

인프라 로그

인프라 네임스페이스에서 생성된 컨테이너 로그: **openshift***, **kube*** 또는 **default** 및 노드의 **journald** 메시지입니다.

감사 로그

/var/log/audit/audit.log 파일에 저장되는 노드 감사 시스템인 **auditd**에서 생성된 로그와 **auditd,kube-apiserver,openshift-apiserver** 서비스 및 활성화된 경우 **ovn** 프로젝트의 로그입니다.

추가 리소스

- [웹 콘솔을 사용한 로그 시작화](#)

4.2. 로깅 배포 정보

관리자는 OpenShift Container Platform 웹 콘솔 또는 OpenShift CLI(oc)를 사용하여 로깅 Operator를 설치하여 로깅을 배포할 수 있습니다. Operator는 로깅의 배포, 업그레이드 및 유지보수를 담당합니다.

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

4.2.1. 사용자 정의 리소스 로깅

각 Operator에서 구현하는 CR(사용자 정의 리소스) YAML 파일을 사용하여 로깅 배포를 구성할 수 있습니다.

Red Hat OpenShift Logging Operator:

- **ClusterLogging (CL) - Operator**가 설치된 후 ClusterLogging 사용자 정의 리소스(CR)를 생성하여 로깅 Pod 및 로깅 지원에 필요한 기타 리소스를 예약합니다. ClusterLogging CR은 수집기 및 전달자를 배포합니다. 현재 각 노드에서 실행되는 데몬 세트로 둘 다 구현됩니다. Red Hat OpenShift Logging Operator는 ClusterLogging CR을 감시하고 그에 따라 로깅 배포를 조정합니다.
- **ClusterLogForwarder (CLF)** - 사용자 구성별로 로그를 전달하도록 수집기 구성을 생성합니다.

ScanSetting Operator:

- **MellanoxStack - CloudEvent** 클러스터를 로그 저장소로 제어하고 **OpenShift Container Platform** 인증 통합을 통해 웹 프록시를 제어하여 멀티 테넌시를 적용합니다.

OpenShift Elasticsearch Operator:



참고

이러한 CR은 **OpenShift Elasticsearch Operator**에서 생성하고 관리합니다.
Operator에서 덮어쓰지 않고 수동 변경을 수행할 수 없습니다.

- **Elasticsearch - Elasticsearch** 인스턴스를 기본 로그 저장소로 구성하고 배포합니다.
- **Kibana** - 로그를 검색, 쿼리 및 볼 수 있도록 **Kibana** 인스턴스를 구성하고 배포합니다.

4.2.2. JSON OpenShift Container Platform 로깅 정보

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

- **JSON** 로그 구문 분석
- **Elasticsearch**의 **JSON** 로그 데이터 구성
- **Elasticsearch** 로그 저장소로 **JSON** 로그 전달

4.2.3. Kubernetes 이벤트 수집 및 저장 정보

OpenShift Container Platform 이벤트 라우터는 **Kubernetes** 이벤트를 감시하고 **OpenShift Container Platform** 로깅의 수집을 위해 해당 이벤트를 기록하는 **Pod**입니다. 이벤트 라우터를 수동으로

배포해야 합니다.

자세한 내용은 [Kubernetes 이벤트 수집 및 저장을](#) 참조하십시오.

4.2.4. 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

4.2.5. 필드 내보내기 정보

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

자세한 내용은 [내보내기 필드](#) 정보를 참조하십시오.

4.2.6. 이벤트 라우팅 정보

이벤트 라우터는 로깅을 통해 수집할 수 있도록 OpenShift Container Platform 이벤트를 감시하는 Pod입니다. 이벤트 라우터는 모든 프로젝트에서 이벤트를 수집하여 STDOUT에 씁니다. Fluentd는 이러한 이벤트를 수집하여 OpenShift Container Platform Elasticsearch 인스턴스로 전달합니다. Elasticsearch는 이벤트를 인프라 인덱스에 인덱싱합니다.

이벤트 라우터를 수동으로 배포해야 합니다.

자세한 내용은 [Kubernetes 이벤트 수집 및 저장을 참조하십시오.](#)

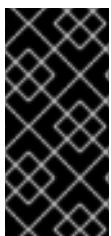
5장. 로깅 설치

Red Hat OpenShift Logging Operator를 설치하여 로깅을 배포할 수 있습니다. **Red Hat OpenShift Logging Operator**는 로깅 스택의 구성 요소를 생성하고 관리합니다.



참고

로깅은 핵심 **OpenShift Container Platform**과 별도의 릴리스 주기와 함께 설치 가능한 구성 요소로 제공됩니다. [Red Hat OpenShift Container Platform](#) 라이프 사이클 정책은 릴리스 호환성에 대해 간단히 설명합니다.



중요

새 설치의 경우 **Vector** 및 **LokiStack**을 사용합니다. **Elasticsearch** 및 **Fluentd**는 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다.

5.1. 웹 콘솔을 사용하여 RED HAT OPENSHIFT LOGGING OPERATOR 설치

OpenShift Container Platform 웹 콘솔을 사용하여 **Red Hat OpenShift Logging Operator**를 설치할 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- **OpenShift Container Platform** 웹 콘솔에 액세스할 수 있습니다.

프로세스

1. **OpenShift Container Platform** 웹 콘솔에서 **Operator** → **OperatorHub**를 클릭합니다.
2. 키워드로 필터링 상자에 **OpenShift Logging** 을 입력합니다.
3. 사용 가능한 **Operator** 목록에서 **Red Hat OpenShift Logging**을 선택한 다음 설치를 클릭합니다.

니다.

4.

설치 모드에서 클러스터의 특정 네임스페이스가 선택되어 있는지 확인합니다.

5.

설치된 네임스페이스에서 **Operator** 권장 네임스페이스가 **openshift-logging**인지 확인하십시오.

6.

이 네임스페이스에서 **Operator** 권장 클러스터 모니터링 사용을 선택합니다.

이 옵션은 **Namespace** 오브젝트에서 **openshift.io/cluster-monitoring: "true"** 라벨을 설정합니다. 클러스터 모니터링이 **openshift-logging** 네임스페이스를 스크랩하도록 하려면 이 옵션을 선택해야 합니다.

7.

stable-5.y 를 업데이트 채널로 선택합니다.



참고

stable 채널은 최신 로깅 릴리스에 대한 업데이트만 제공합니다. 이전 릴리스에 대한 업데이트를 계속 받으려면 서브스크립션 채널을 **stable-x.y**로 변경해야 합니다. 여기서 **x.y**는 설치한 로깅 및 마이너 버전을 나타냅니다. 예를 들면 **stable-5.7**입니다.

8.

업데이트 승인을 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.

9.

콘솔 플러그인에 대해 활성화 또는 비활성화를 선택합니다.

10.

설치를 클릭합니다.

검증

1.

Operator → 설치된 **Operator** 페이지로 전환하여 **Red Hat OpenShift Logging Operator** 가 설치되었는지 확인합니다.

2.

상태 열에서 **InstallSucceeded** 가 포함된 녹색 확인 표시와 최대 날짜 텍스트가 표시되는지 확인합니다.



중요

Operator는 설치가 완료되기 전에 실패 상태를 표시할 수 있습니다. **Operator** 설치가 **InstallSucceeded** 메시지와 함께 완료되면 페이지를 새로 고칩니다.

Operator가 설치된 것으로 표시되지 않으면 다음 문제 해결 옵션 중 하나를 선택합니다.

-

Operator → 설치된 **Operator** 페이지로 이동하여 상태 열에 오류 또는 실패가 있는지 검사합니다.

-

워크로드 → **Pod** 페이지로 이동하여 **openshift-logging** 프로젝트에서 문제를 보고하는 **Pod** 의 로그를 확인합니다.

5.2. 웹 콘솔을 사용하여 CLUSTERLOGGING 오브젝트 생성

로깅 **Operator**를 설치한 후 클러스터의 로그 스토리지, 시각화 및 로그 수집기를 구성하려면 **ClusterLogging** 사용자 정의 리소스를 생성해야 합니다.

사전 요구 사항

-

Red Hat OpenShift Logging Operator가 설치되어 있습니다.

-

OpenShift Container Platform 웹 콘솔 관리자 화면에 액세스할 수 있습니다.

프로세스

1. **Custom Resource Definitions** 페이지로 이동합니다.
2. 사용자 정의 리소스 정의 페이지에서 **ClusterLogging**을 클릭합니다.
3. 사용자 정의 리소스 정의 상세 정보 페이지의 작업 메뉴에서 인스턴스 보기 선택합니다.
4. **ClusterLoggings** 페이지에서 **ClusterLogging** 생성을 클릭합니다.
5. 컬렉션 섹션에서 수집기 구현을 선택합니다.



참고

Fluentd는 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다. **Red Hat**은 현재 릴리스 라이프사이클 동안 이 기능에 대한 버그 수정 및 지원을 제공하지만 이 기능은 더 이상 개선 사항을 받지 않습니다. **Fluentd** 대신 **Vector**를 사용할 수 있습니다.

- 6.

logStore 섹션에서 유형을 선택합니다.



참고

로깅 5.9 릴리스에는 업데이트된 **OpenShift Elasticsearch Operator** 버전이 포함되어 있지 않습니다. 현재 **Logging 5.8**과 함께 릴리스된 **OpenShift Elasticsearch Operator**를 사용하는 경우 로깅 5.8의 EOL까지 로깅에서 계속 작동합니다. **OpenShift Elasticsearch Operator**를 사용하여 기본 로그 스토리지를 관리하는 대신 **Loki Operator**를 사용할 수 있습니다. 로깅 라이프사이클 날짜에 대한 자세한 내용은 [Platform Agnostic Operators](#)를 참조하십시오.

7.

생성을 클릭합니다.

5.3. CLI를 사용하여 RED HAT OPENSHIFT LOGGING OPERATOR 설치

OpenShift CLI(oc)를 사용하여 Red Hat OpenShift Logging Operator를 설치할 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- OpenShift CLI(oc)가 설치되어 있습니다.

프로세스

1. Namespace 오브젝트를 YAML 파일로 생성합니다.

Namespace 오브젝트의 예

```
apiVersion: v1
kind: Namespace
metadata:
  name: <name> ①
  annotations:
    openshift.io/node-selector: ""
  labels:
    openshift.io/cluster-monitoring: "true"
```

①

로깅 버전 5.7 및 이전 버전의 네임스페이스 이름으로 **openshift-logging** 을 지정해야 합니다. 로깅 5.8 이상 버전의 경우 모든 이름을 사용할 수 있습니다.

2.

다음 명령을 실행하여 **Namespace** 오브젝트를 적용합니다.

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

3.

OperatorGroup 오브젝트를 YAML 파일로 생성합니다.

OperatorGroup 오브젝트의 예

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

① ②

로깅 버전 5.7 이상에는 **openshift-logging** 네임스페이스를 지정해야 합니다. 로깅 5.8 이상 버전의 경우 모든 네임스페이스를 사용할 수 있습니다.

4.

다음 명령을 실행하여 **OperatorGroup** 오브젝트를 적용합니다.

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

5.

Red Hat OpenShift Logging Operator에 네임스페이스를 서브스크립션하는 **Subscription** 오브젝트를 생성합니다.

Subscription 개체 예

```
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

로깅 버전 5.7 이상에는 **openshift-logging** 네임스페이스를 지정해야 합니다. 로깅 5.8 이상 버전의 경우 모든 네임스페이스를 사용할 수 있습니다.

2

stable 또는 **stable-x.y** 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.

6.

다음 명령을 실행하여 서브스크립션을 적용합니다.

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

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

검증

1.

다음 명령을 실행합니다.

```
$ oc get csv -n <namespace>
```

2.

출력을 관찰하고 네임스페이스에 Red Hat OpenShift Logging Operator가 있는지 확인합니다.

출력 예

NAMESPACE	VERSION	REPLACES	NAME	DISPLAY
		PHASE		
...				
openshift-logging			clusterlogging.5.8.0-202007012112.p0	
OpenShift Logging		5.8.0-202007012112.p0		Succeeded
...				

5.4. CLI를 사용하여 CLUSTERLOGGING 오브젝트 생성

이 기본 로깅 구성은 다양한 환경을 지원합니다. 수행할 수 있는 수정 사항에 대한 정보는 구성 요소 튜닝 및 구성 주제를 검토하십시오.

사전 요구 사항

- Red Hat OpenShift Logging Operator가 설치되어 있습니다.
- 로그 저장소에 대해 OpenShift Elasticsearch Operator를 설치했습니다.
- OpenShift CLI(oc)가 설치되어 있습니다.

프로세스

- ClusterLogging 오브젝트를 YAML 파일로 생성합니다.

ClusterLogging 오브젝트의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
  name: instance ①
  namespace: openshift-logging
```

```

spec:
  managementState: Managed ②
  logStore:
    type: elasticsearch ③
    retentionPolicy:
      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:
    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 Logging`이 다시 `Managed` 상태로 될 때까지 업데이트를 받지 않습니다.

3

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 7d입니다. **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 기준 스토리지 클래스 이름을 입력합니다. 최상의 성능을 위해서는 블록 스토리지를 할당하는 스토리지 클래스를 지정합니다. 스토리지 클래스를 지정하지 않으면 **OpenShift Logging**은 임시 스토리지를 사용합니다.

7

필요에 따라 **Elasticsearch**에 대한 CPU 및 메모리 요청을 지정합니다. 이 값을 비워 두면 **OpenShift Elasticsearch Operator**가 대부분의 배포에 충분한 기본값으로 설정합니다. 기본값은 메모리 요청 시 16Gi이고 CPU 요청 시 1입니다.

8

필요에 따라 **Elasticsearch** 프록시에 대한 CPU 및 메모리 요청을 지정합니다. 이 값을 비워 두면 **OpenShift Elasticsearch Operator**가 대부분의 배포에 충분한 기본값으로 설정합니다. 기본값은 메모리 요청 시 256Mi이고 CPU 요청 시 100m입니다.

9

Kibana 구성을 위한 설정입니다. CR을 사용하여 중복성을 위해 **Kibana**를 확장하고 **Kibana** 노드의 CPU 및 메모리를 구성할 수 있습니다. 자세한 내용은 로그 시각화 프로그램 구성을 참조하십시오.

10

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

참고

Elasticsearch 컨트롤 플레인 노드의 최대 수는 3입니다. 3보다 큰 **nodeCount**를 지정하면 **OpenShift Container Platform**은 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 **shard**를 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

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
cluster-logging-operator	1/1	1	1	18h
elasticsearch-cd-x6kdekli-1	1/1	1	1	6m54s
elasticsearch-cdm-x6kdekli-1	1/1	1	1	18h
elasticsearch-cdm-x6kdekli-2	1/1	1	1	6m49s
elasticsearch-cdm-x6kdekli-3	1/1	1	1	6m44s

인덱스 템플릿의 기본 **shard** 수는 **Elasticsearch** 데이터 노드 수와 같습니다.

검증

openshift-logging 프로젝트에 **Pod**를 나열하여 설치를 확인할 수 있습니다.

- 다음 명령을 실행하여 **Pod**를 나열합니다.

\$ oc get pods -n openshift-logging

다음 목록과 유사하게 로깅 구성 요소의 Pod를 관찰합니다.

출력 예

NAME	READY	STATUS	RESTARTS	AGE
cluster-logging-operator-66f77ffccb-ppzbg	1/1	Running	0	7m
elasticsearch-cdm-ftuhduuw-1-ffc4b9566-q6bhp	2/2	Running	0	2m40s
elasticsearch-cdm-ftuhduuw-2-7b4994dbfc-rd2gc	2/2	Running	0	2m36s
elasticsearch-cdm-ftuhduuw-3-84b5ff7ff8-gqnm2	2/2	Running	0	2m4s
collector-587vb	1/1	Running	0	2m26s
collector-7mpb9	1/1	Running	0	2m30s
collector-flm6j	1/1	Running	0	2m33s
collector-gn4rn	1/1	Running	0	2m26s
collector-nlgb6	1/1	Running	0	2m30s
collector-snptk	1/1	Running	0	2m28s
kibana-d6d5668c5-rppqm	2/2	Running	0	2m39s

5.5. 설치 후 작업

Red Hat OpenShift Logging Operator를 설치한 후 ClusterLogging 사용자 정의 리소스(CR)를 생성하고 수정하여 배포를 구성할 수 있습니다.

작은 정보

Elasticsearch 로그 저장소를 사용하지 않는 경우 ClusterLogging 사용자 정의 리소스(CR)에서 내부 Elasticsearch logStore 및 Kibana visualization 구성 요소를 제거할 수 있습니다. 이러한 구성 요소를 제거하는 것은 선택 사항이지만 리소스를 절약할 수 있습니다. [Elasticsearch 로그 저장소를 사용하지 않는 경우 사용되지 않는 구성 요소 제거를 참조하십시오.](#)

5.5.1. 클러스터 로깅 사용자 정의 리소스 정보

로깅 환경을 변경 하려면 ClusterLogging 사용자 정의 리소스(CR)를 생성하고 수정합니다.

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
# ...

```

1

CR 이름은 **instance**여야 합니다.

2

CR은 **openshift-logging** 네임스페이스에 설치해야 합니다.

3

Red Hat OpenShift Logging Operator 관리 상태입니다. 상태가 **Unmanaged**로 설정된 경우 **Operator**는 지원되지 않는 상태에 있으며 업데이트가 제공되지 않습니다.

5.5.2. 로그 스토리지 구성

ClusterLogging 사용자 정의 리소스(CR)를 수정하여 로깅에서 사용하는 로그 스토리지 유형을 구성할 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- **OpenShift CLI(oc)**가 설치되어 있습니다.
- **Red Hat OpenShift Logging Operator**와 **LokiStack** 또는 **Elasticsearch**인 내부 로그 저장소를 설치했습니다.

- ClusterLogging CR을 생성했습니다.



참고

로깅 5.9 릴리스에는 업데이트된 **OpenShift Elasticsearch Operator** 버전이 포함되어 있지 않습니다. 현재 **Logging 5.8**과 함께 릴리스된 **OpenShift Elasticsearch Operator**를 사용하는 경우 로깅 5.8의 EOL까지 로깅에서 계속 작동합니다. **OpenShift Elasticsearch Operator**를 사용하여 기본 로그 스토리지를 관리하는 대신 **Loki Operator**를 사용할 수 있습니다. 로깅 라이프사이클 날짜에 대한 자세한 내용은 [Platform Agnostic Operators](#) 를 참조하십시오.

프로세스

1.

- ClusterLogging CR logStore 사양을 수정합니다.

ClusterLogging CR 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
# ...
spec:
# ...
logStore:
  type: <log_store_type> ①
  elasticsearch: ②
    nodeCount: <integer>
    resources: {}
    storage: {}
  redundancyPolicy: <redundancy_type> ③
  lokistack: ④
    name: {}
# ...
```

①

로그 저장소 유형을 지정합니다. **lokistack** 또는 **elasticsearch** 일 수 있습니다.

②

③

3

중복 유형을 지정합니다. 이 값은 **ZeroRedundancy**, **SingleRedundancy**, **MultipleRedundancy** 또는 **FullRedundancy** 일 수 있습니다.

4

LokiStack에 대한 선택적 구성 옵션입니다.

LokiStack을 로그 저장소로 지정하는 **ClusterLogging CR**의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
  name: instance
  namespace: openshift-logging
spec:
  managementState: Managed
  logStore:
    type: lokistack
    lokistack:
      name: logging-loki
# ...
```

2.

다음 명령을 실행하여 **ClusterLogging CR**을 적용합니다.

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

5.5.3. 로그 수집기 구성

ClusterLogging 사용자 정의 리소스(**CR**)를 수정하여 로깅에서 사용하는 로그 수집기 유형을 구성할 수 있습니다.



참고

Fluentd는 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다. **Red Hat**은 현재 릴리스 라이프사이클 동안 이 기능에 대한 버그 수정 및 지원을 제공하지만 이 기능은 더 이상 개선 사항을 받지 않습니다. **Fluentd** 대신 **Vector**를 사용할 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- **OpenShift CLI(oc)**가 설치되어 있습니다.
- **Red Hat OpenShift Logging Operator**가 설치되어 있습니다.
- **ClusterLogging CR**을 생성했습니다.

프로세스

1. **ClusterLogging CR** 컬렉션 사양을 수정합니다.

ClusterLogging CR 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
# ...
spec:
# ...
collection:
  type: <log_collector_type> ①
  resources: {}
  tolerations: {}
# ...
```

2.

다음 명령을 실행하여 **ClusterLogging CR**을 적용합니다.

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

5.5.4. 로그 시각화 프로그램 구성

ClusterLogging 사용자 정의 리소스(**CR**)를 수정하여 로깅에서 사용하는 로그 시각화 프로그램 유형을 구성할 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- OpenShift CLI(oc)**가 설치되어 있습니다.
- Red Hat OpenShift Logging Operator**가 설치되어 있습니다.
- ClusterLogging CR**을 생성했습니다.



중요

시각화에 **OpenShift Container Platform** 웹 콘솔을 사용하려면 로깅 콘솔 플러그인을 활성화해야 합니다. "웹 콘솔을 사용한 로그 시각화"에 대한 설명서를 참조하십시오.

프로세스

1.

ClusterLogging CR 시각화 사양을 수정합니다.

ClusterLogging CR 예

```

apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
# ...
spec:
# ...
visualization:
  type: <visualizer_type> ①
  kibana: ②
  resources: {}
  nodeSelector: {}
  proxy: {}
  replicas: {}
  tolerations: {}
  ocpConsole: ③
  logsLimit: {}
  timeout: {}
# ...

```

①

로깅에 사용할 시각화 프로그램 유형입니다. **kibana** 또는 **ocp-console** 일 수 있습니다. **Kibana** 콘솔은 **Elasticsearch** 로그 스토리지를 사용하는 배포와만 호환되며 **OpenShift Container Platform** 콘솔은 **LokiStack** 배포와만 호환됩니다.

②

Kibana 콘솔의 선택적 구성입니다.

③

OpenShift Container Platform 웹 콘솔의 선택적 구성입니다.

2.

다음 명령을 실행하여 **ClusterLogging CR**을 적용합니다.

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

5.5.5. 네트워크 분리가 활성화될 때 프로젝트 간 트래픽 허용

클러스터 네트워크 플러그인에서 네트워크 분리를 적용할 수 있습니다. 이 경우 **OpenShift Logging**에서 배포한 **operator**가 포함된 프로젝트 간 네트워크 트래픽을 허용해야 합니다.

네트워크 분리는 다른 프로젝트에 있는 pod 또는 서비스 간의 네트워크 트래픽을 차단합니다. 로깅은 **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 네트워크 플러그인인 **OpenShift SDN** 및 **OVN-Kubernetes**에 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 SDN**과 마찬가지로 하나의 로깅 관련 프로젝트에서 다른 프로젝트로 트래픽이 송신될 수 있도록 정책을 구성해야 합니다.

프로세스

- 다중 테넌트 모드에서 **OpenShift SDN**을 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-Kubernetes**의 **OpenShift SDN**의 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 생성합니다. 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](#)

6장. 로깅 업데이트

로깅 업데이트에는 마이너 릴리스 업데이트(**5.y.z**) 및 주요 릴리스 업데이트(**5.y**)의 두 가지 유형이 있습니다.

6.1. 마이너 릴리스 업데이트

자동 업데이트 승인 옵션을 사용하여 로깅 **Operator**를 설치한 경우 **Operator**에 마이너 버전 업데이트가 자동으로 제공됩니다. 수동 업데이트 단계를 완료할 필요가 없습니다.

수동 업데이트 승인 옵션을 사용하여 로깅 **Operator**를 설치한 경우 마이너 버전 업데이트를 수동으로 승인해야 합니다.

자세한 내용은 [보류 중인 Operator 업데이트 수동 승인](#) 을 참조하십시오.

6.2. 주요 릴리스 업데이트

주요 버전 업데이트의 경우 일부 수동 단계를 완료해야 합니다.

주요 릴리스 버전 호환성 및 지원 정보는 [OpenShift Operator 라이프 사이클](#) 을 참조하십시오.

6.3. 모든 네임스페이스를 조사하도록 RED HAT OPENSHIFT LOGGING OPERATOR 업그레이드

로깅 5.7 이전 버전에서는 Red Hat OpenShift Logging Operator는 `openshift-logging` 네임스페이스만 감시합니다. Red Hat OpenShift Logging Operator에서 클러스터의 모든 네임스페이스를 조사하려면 **Operator**를 재배포해야 합니다. 로깅 구성 요소를 삭제하지 않고 **Operator**를 재배포하려면 다음 절차를 완료할 수 있습니다.

사전 요구 사항

- **OpenShift CLI(oc)**가 설치되어 있습니다.
- 관리자 권한이 있습니다.

프로세스

1. 다음 명령을 실행하여 서브스크립션을 삭제합니다.

```
$ oc -n openshift-logging delete subscription <subscription>
```

2. 다음 명령을 실행하여 Operator group을 삭제합니다.

```
$ oc -n openshift-logging delete operatorgroup <operator_group_name>
```

3. 다음 명령을 실행하여 CSV(클러스터 서비스 버전)를 삭제합니다.

```
$ oc delete clusterserviceversion cluster-logging.<version>
```

4. "로깅 설치" 설명서에 따라 Red Hat OpenShift Logging Operator를 재배포합니다.

검증

- OperatorGroup 리소스의 targetNamespaces 필드가 없거나 빈 문자열로 설정되어 있는지 확인합니다.

이렇게 하려면 다음 명령을 실행하고 출력을 검사합니다.

```
$ oc get operatorgroup <operator_group_name> -o yaml
```

출력 예

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: openshift-logging-f52cn
  namespace: openshift-logging
spec:
  upgradeStrategy: Default
status:
  namespaces:
  - ""
# ...
```

6.4. RED HAT OPENSHIFT LOGGING OPERATOR 업데이트

Red Hat OpenShift Logging Operator를 새 주요 릴리스 버전으로 업데이트하려면 **Operator** 서비스 크립션의 업데이트 채널을 수정해야 합니다.

사전 요구 사항

- **Red Hat OpenShift Logging Operator**가 설치되어 있습니다.
- 관리자 권한이 있습니다.
- **OpenShift Container Platform** 웹 콘솔에 액세스하고 관리자 화면을 보고 있습니다.

프로세스

1. **Operators** → 설치된 **Operator**로 이동합니다.
2. **openshift-logging** 프로젝트를 선택합니다.
3. **Red Hat OpenShift Logging Operator**를 클릭합니다.
4. 서브스크립션을 클릭합니다. 서브스크립션 세부 정보 섹션에서 채널 업데이트 링크를 클릭합니다. 이 링크 텍스트는 현재 업데이트 채널에 따라 **stable** 또는 **stable-5.y** 일 수 있습니다.
5. 서브스크립션 업데이트 채널 변경 창에서 최신 주요 버전 업데이트 채널, **stable-5.y**를 선택하고 저장을 클릭합니다. **cluster-logging.v5.y.z** 버전을 확인합니다.

검증

- 1.

몇 초 정도 기다린 후 **Operator** → 설치된 **Operator**를 클릭합니다. **Red Hat OpenShift Logging Operator** 버전이 최신 `cluster-logging.v5.y.z` 버전과 일치하는지 확인합니다.

2.

Operator → 설치된 **Operator** 페이지에서 **Status** 필드가 성공으로 표시될 때까지 기다립니다.

6.5. LOKI OPERATOR 업데이트

Loki Operator를 새 주요 릴리스 버전으로 업데이트하려면 **Operator** 서브스크립션의 업데이트 채널을 수정해야 합니다.

사전 요구 사항

- **Loki Operator**를 설치했습니다.
- 관리자 권한이 있습니다.
- **OpenShift Container Platform** 웹 콘솔에 액세스하고 관리자 화면을 보고 있습니다.

프로세스

1.

Operators → 설치된 **Operator**로 이동합니다.

2.

openshift-operators-redhat 프로젝트를 선택합니다.

3.

Loki Operator를 클릭합니다.

4.

서브스크립션을 클릭합니다. 서브스크립션 세부 정보 섹션에서 채널 업데이트 링크를 클릭합니다. 이 링크 텍스트는 현재 업데이트 채널에 따라 **stable** 또는 **stable-5.y** 일 수 있습니다.

5.

서브스크립션 업데이트 채널 변경 창에서 최신 주요 버전 업데이트 채널, **stable-5.y**를 선택하고 저장을 클릭합니다. **loki-operator.v5.y.z** 버전을 확인합니다.

검증

1. 몇 초 정도 기다린 후 **Operator** → 설치된 **Operator**를 클릭합니다. **Loki Operator** 버전이 최신 **loki-operator.v5.y.z** 버전과 일치하는지 확인합니다.
2. **Operator** → 설치된 **Operator** 페이지에서 **Status** 필드가 성공으로 표시될 때까지 기다립니다.

6.6. OPENSHIFT ELASTICSEARCH OPERATOR 업데이트

OpenShift Elasticsearch Operator를 현재 버전으로 업데이트하려면 서브스크립션을 수정해야 합니다.



참고

로깅 5.9 릴리스에는 업데이트된 **OpenShift Elasticsearch Operator** 버전이 포함되어 있지 않습니다. 현재 **Logging 5.8**과 함께 릴리스된 **OpenShift Elasticsearch Operator**를 사용하는 경우 로깅 5.8의 EOL까지 로깅에서 계속 작동합니다. **OpenShift Elasticsearch Operator**를 사용하여 기본 로그 스토리지를 관리하는 대신 **Loki Operator**를 사용할 수 있습니다. 로깅 라이프사이클 날짜에 대한 자세한 내용은 [Platform Agnostic Operators](#)를 참조하십시오.

사전 요구 사항

- Elasticsearch를 기본 로그 저장소로 사용하고 UI로 Kibana를 사용하는 경우 Red Hat OpenShift Logging Operator를 업데이트하기 전에 OpenShift Elasticsearch Operator를 업데이트합니다.



중요

Operator를 잘못된 순서로 업데이트하면 Kibana가 업데이트되지 않고 Kibana 사용자 정의 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.

- 로깅 상태가 정상입니다.

- 모든 Pod의 상태가 ready입니다.
- Elasticsearch 클러스터는 정상입니다.
- Elasticsearch 및 Kibana 데이터가 백업됩니다.
- 관리자 권한이 있습니다.
- 확인 단계를 위해 OpenShift CLI(oc)를 설치했습니다.

절차

1. OpenShift Container Platform 웹 콘솔에서 Operator → 설치된 Operator를 클릭합니다.
2. openshift-operators-redhat 프로젝트를 선택합니다.
3. OpenShift Elasticsearch Operator를 클릭합니다.
4. 서브스크립션 → 채널을 클릭합니다.
5. 서브스크립션 업데이트 채널 변경 창에서 stable-5.y를 선택하고 저장을 클릭합니다.
elasticsearch-operator.v5.y.z 버전을 확인합니다.
6. 몇 초 정도 기다린 후 Operator → 설치된 Operator를 클릭합니다. OpenShift Elasticsearch Operator 버전이 최신 elasticsearch-operator.v5.y.z 버전과 일치하는지 확인합니다.
7. Operator → 설치된 Operator 페이지에서 Status 필드가 성공으로 표시될 때까지 기다립니다.

검증

1.

다음 명령을 입력하고 출력을 관찰하여 모든 **Elasticsearch Pod**의 상태가 **Ready** 인지 확인합니다.

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

출력 예

NAME	READY	STATUS	RESTARTS	AGE
elasticsearch-cdm-1pb1l44l-1-55b7546f4c-mshhk	2/2	Running	0	31m
elasticsearch-cdm-1pb1l44l-2-5c6d87589f-gx5hk	2/2	Running	0	30m
elasticsearch-cdm-1pb1l44l-3-88df5d47-m45jc	2/2	Running	0	29m

2.

다음 명령을 입력하고 출력을 관찰하여 **Elasticsearch** 클러스터 상태가 녹색 인지 확인합니다.

```
$ oc exec -n openshift-logging -c elasticsearch elasticsearch-cdm-1pb1l44l-1-55b7546f4c-mshhk -- health
```

출력 예

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

3.

다음 명령을 입력하고 출력을 관찰하여 **Elasticsearch cron** 작업이 생성되었는지 확인합니다.

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

```
$ oc get cronjob
```

출력 예

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

4.

다음 명령을 입력하고 출력을 관찰하여 로그 저장소가 올바른 버전으로 업데이트되고 인덱스가 녹색인지 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 6.1. 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				
rtDSzoqsSI6saIsSK7Au1Q	3 1	226717	0	297	148	
green	open	infra-000012				
RSf_kUwDSR2xEuKRZMPqZQ	3 1	227623	0	295	147	
green	open	.kibana_7				
1SJdCqlZTPWIIAaOUD78yg	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_GVQVZ6Yh9lg	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-		

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	
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						
RVp7TemSSemGJcsSUmf3A	1	1	4	0	0	0
green open infra-000011						
J7XWBauWSTe0jnZX02fU6A	3	1	226100	0	293	146
green open app-000001						
axSAFfONQDmKwatkjPXdtw	3	1	103186	0	126	57
green open infra-000016						
m9c1iRLtStWSF1GopaRyCg	3	1	13685	0	19	9
green open infra-000002						Hz6WvINtTvKcQzw-
ewmbYg	3	1	228994	0	296	148
green open infra-000013						KR9mMFUpQI-
jraYtanylGw	3	1	228166	0	298	148
green open audit-000001						
eERqLdLmQOjQDFES1LBATQ	3	1		0	0	0

5.

다음 명령을 입력하고 출력을 관찰하여 로그 시각화 프로그램이 올바른 버전으로 업데이트되었는지 확인합니다.

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

출력에 **ready** 상태가 있는 **Kibana pod**가 포함되어 있는지 확인합니다.

예 6.2. 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
}
```

7장. 로그 시각화

7.1. 로그 시각화 정보

배포된 로그 스토리지 솔루션에 따라 OpenShift Container Platform 웹 콘솔 또는 Kibana 웹 콘솔에서 로그 데이터를 시각화할 수 있습니다. Kibana 콘솔은 Elasticsearch 로그 저장소와 함께 사용할 수 있으며 OpenShift Container Platform 웹 콘솔은 Elasticsearch 로그 저장소 또는 LokiStack과 함께 사용할 수 있습니다.



참고

Kibana 웹 콘솔은 향후 로깅 릴리스에서 더 이상 사용되지 않습니다.

7.1.1. 로그 시각화 프로그램 구성

ClusterLogging 사용자 정의 리소스(CR)를 수정하여 로깅에서 사용하는 로그 시각화 프로그램 유형을 구성할 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- OpenShift CLI(oc)가 설치되어 있습니다.
- Red Hat OpenShift Logging Operator가 설치되어 있습니다.
- ClusterLogging CR을 생성했습니다.



중요

시각화에 OpenShift Container Platform 웹 콘솔을 사용하려면 로깅 콘솔 플러그인을 활성화해야 합니다. "웹 콘솔을 사용한 로그 시각화"에 대한 설명서를 참조하십시오.

절차

- ClusterLogging CR 시작화 사양을 수정합니다.

ClusterLogging CR 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
# ...
spec:
# ...
  visualization:
    type: <visualizer_type> ①
    kibana: ②
      resources: {}
      nodeSelector: {}
      proxy: {}
      replicas: {}
      tolerations: {}
    ocpConsole: ③
      logsLimit: {}
      timeout: {}
# ...
```

1

로깅에 사용할 시작화 프로그램 유형입니다. **kibana** 또는 **ocp-console** 일 수 있습니다. Kibana 콘솔은 **Elasticsearch** 로그 스토리지를 사용하는 배포와만 호환되며 OpenShift Container Platform 콘솔은 **LokiStack** 배포와만 호환됩니다.

2

Kibana 콘솔의 선택적 구성입니다.

3

OpenShift Container Platform 웹 콘솔의 선택적 구성입니다.

- 다음 명령을 실행하여 ClusterLogging CR을 적용합니다.

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

7.1.2. 리소스의 로그 보기

리소스 로그는 제한된 로그 보기 기능을 제공하는 기본 기능입니다. **OpenShift CLI(oc)** 및 웹 콘솔을 사용하여 빌드, 배포 및 **Pod**와 같은 다양한 리소스의 로그를 볼 수 있습니다.

작은 정보

로그 검색 및 보기 환경을 개선하려면 로깅을 설치합니다. 로깅은 노드 시스템 감사 로그, 애플리케이션 컨테이너 로그 및 인프라 로그와 같은 **OpenShift Container Platform** 클러스터의 모든 로그를 전용 로그 저장소로 집계합니다. 그런 다음 **Kibana** 콘솔 또는 **OpenShift Container Platform** 웹 콘솔을 통해 로그 데이터를 쿼리, 검색 및 시각화할 수 있습니다. 리소스 로그는 로깅 로그 저장소에 액세스하지 않습니다.

7.1.2.1. 리소스 로그 보기

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

사전 요구 사항

- **OpenShift CLI(oc)**에 액세스합니다.

프로세스(UI)

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



참고

빌드와 같은 일부 리소스에는 직접 쿼리할 **Pod**가 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: 출력에서 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.

7.2. 웹 콘솔을 사용한 로그 시각화

OpenShift Container Platform 웹 콘솔을 사용하여 로깅 콘솔 플러그인을 구성하여 로그 데이터를 시각화할 수 있습니다.

로깅 설치 중에 플러그인을 구성하는 방법에 대한 자세한 내용은 [웹 콘솔을 사용하여 로깅 설치를 참조하십시오.](#)

이미 로깅을 설치하고 플러그인을 구성하려면 다음 절차 중 하나를 사용합니다.

7.2.1. Red Hat OpenShift Logging Operator를 설치한 후 로깅 콘솔 플러그인 활성화

Red Hat OpenShift Logging Operator 설치의 일부로 로깅 콘솔 플러그인을 활성화할 수 있지만 플러그인이 비활성화된 상태에서 **Red Hat OpenShift Logging Operator**를 이미 설치한 경우 플러그인을 활성화할 수도 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- Red Hat OpenShift Logging Operator**를 설치하고 **Console** 플러그인에 대해 **Disabled**를 선택했습니다.

-

OpenShift Container Platform 웹 콘솔에 액세스할 수 있습니다.

절차

1. **OpenShift Container Platform** 웹 콘솔 관리자 화면에서 **Operator** → 설치된 **Operator**로 이동합니다.
2. **Red Hat OpenShift Logging** 을 클릭합니다. 그러면 **Operator** 세부 정보 페이지로 이동합니다.
3. 세부 정보 페이지에서 **Console** 플러그인 옵션에 대해 **Disabled**를 클릭합니다.
4. 콘솔 플러그인 활성화 대화 상자에서 **Enable**을 선택합니다.
5. 저장을 클릭합니다.
6. **Console** 플러그인 옵션에 **Enabled**가 표시되는지 확인합니다.
7. 변경 사항이 적용되면 웹 콘솔에 팝업 창이 표시됩니다. 웹 콘솔을 다시 로드하라는 창이 표시됩니다. 팝업 창이 표시되면 브라우저를 새로 고침하여 변경 사항을 적용합니다.

7.2.2. Elasticsearch 로그 저장소 및 LokiStack이 설치된 경우 로깅 콘솔 플러그인 구성

로깅 버전 5.8 이상에서는 **Elasticsearch** 로그 저장소가 기본 로그 저장소이지만 **LokiStack**도 설치한 경우 다음 절차를 사용하여 로깅 콘솔 플러그인을 활성화할 수 있습니다.

사전 요구 사항

-

관리자 권한이 있습니다.

-

Red Hat OpenShift Logging Operator, **OpenShift Elasticsearch Operator** 및 **Loki Operator**를 설치했습니다.

- OpenShift CLI(oc)가 설치되어 있습니다.
- ClusterLogging 사용자 정의 리소스(CR)를 생성했습니다.

절차

1.

다음 명령을 실행하여 로깅 콘솔 플러그인이 활성화되어 있는지 확인합니다.

```
$ oc get consoles.operator.openshift.io cluster -o yaml |grep logging-view-plugin \
|| oc patch consoles.operator.openshift.io cluster --type=merge \
--patch '{ "spec": { "plugins": ["logging-view-plugin"]}}'
```

2.

다음 명령을 실행하여 ClusterLogging CR에
.metadata.annotations.logging.openshift.io/ocp-console-migration-target: lokistack-dev
주석을 추가합니다.

```
$ oc patch clusterlogging instance --type=merge --patch \
'{ "metadata": { "annotations": { "logging.openshift.io/ocp-console-migration-target": \
"lokistack-dev" }}}' \
-n openshift-logging
```

출력 예

```
clusterlogging.logging.openshift.io/instance patched
```

검증

•

다음 명령을 실행하고 출력을 관찰하여 주석이 성공적으로 추가되었는지 확인합니다.

```
$ oc get clusterlogging instance \
-o=jsonpath='{.metadata.annotations.logging\\.openshift\\.io/ocp-console-migration- \
target}' \
-n openshift-logging
```

출력 예

"lokistack-dev"

이제 로깅 콘솔 플러그인 포드가 배포됩니다. **OpenShift Container Platform** 웹 콘솔로 이동하여 모니터링 → 로그 페이지를 확인하여 로깅 데이터를 볼 수 있습니다.

7.3. 클러스터 대시보드 보기

OpenShift Container Platform 웹 콘솔의 로깅/Elasticsearch 노드 및 Openshift 로깅 대시보드에는 문제를 방지하고 진단하는 데 사용할 수 있는 Elasticsearch 인스턴스 및 개별 Elasticsearch 노드에 대한 자세한 정보가 포함되어 있습니다.

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.

7.3.1. Elasticsearch 및 OpenShift Logging 대시보드에 액세스

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

절차

대시보드를 시작하려면 다음을 수행합니다.

1.

OpenShift Container Platform 웹 콘솔에서 **Observe** → **Dashboards**를 클릭합니다.

2.

대시보드 페이지의 대시보드 메뉴에서 로깅/Elasticsearch 노드 또는 OpenShift Logging 을 선택합니다.

로깅/Elasticsearch 노드 대시보드의 경우 보려는 Elasticsearch 노드를 선택하고 데이터 해상도를 설정할 수 있습니다.

여러 데이터 차트를 보여주는 적절한 대시보드가 표시됩니다.

3.

선택 사항: 시간 범위 및 새로 고침 간격 메뉴에서 데이터를 표시하거나 새로 고침 다른 시간 범위를 선택합니다.

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

7.3.2. OpenShift 로깅 대시보드 정보

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

표 7.1. OpenShift 로깅 차트

지표	설명
Elastic 클러스터 상태	현재 Elasticsearch 상태: <ul style="list-style-type: none"> 온라인 - Elasticsearch 인스턴스가 온라인 상태임을 나타냅니다. 오프라인 - Elasticsearch 인스턴스가 오프라인 상태임을 나타냅니다.
Elastic 노드	Elasticsearch 인스턴스의 총 Elasticsearch 노드 수입니다.
Elastic Shard	Elasticsearch 인스턴스의 총 Elasticsearch shard 수입니다.
Elastic 문서	Elasticsearch 인스턴스의 총 Elasticsearch 문서 수입니다.
디스크의 총 인덱스 크기	Elasticsearch 인덱스에 사용 중인 총 디스크 공간입니다.

지표	설명
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"> ● 쿼리 대기 시간: 각 Elasticsearch 검색 쿼리를 실행하는 데 걸리는 평균 시간입니다. ● 가져오기 대기 시간: 각 Elasticsearch 검색 쿼리가 데이터를 가져오는 데 소요되는 평균 시간입니다. <p>가져오기 대기 시간은 일반적으로 쿼리 대기 시간보다 더 깊습니다. 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).
사용된 Elastic JVM 힙	사용된 JVM 메모리 양입니다. 정상 클러스터에서 그래프는 JVM 가비지 수집에 의해 메모리가 해제됨에 따라 정기적으로 감소를 표시합니다.
Elasticsearch 디스크 사용량	각 Elasticsearch 노드에 대해 Elasticsearch 인스턴스에서 사용하는 총 디스크 공간입니다.
사용 중인 파일 설명자	Elasticsearch, Fluentd 및 Kibana에서 사용하는 총 파일 설명자 수입니다.
FluentD 방출 수	FluentD 기본 출력에 대한 초당 총 FluentD 메시지 수 및 기본 출력에 대한 재시도 횟수입니다.
FluentD 버퍼 사용	첨크에 사용되는 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.

지표	설명
FluentD 출력 오류율	FluentD가 로그를 출력할 수 없는 초당 총 횟수입니다.

7.3.3. 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 7.2. Elasticsearch 상태 펠드

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

Elasticsearch 클러스터 인덱스 shard 상태

각 Elasticsearch 인덱스는 지속되는 데이터의 기본 단위인 하나 이상의 shard로 구성된 논리적

그룹입니다. 인덱스 shard는 기본 shard와 복제본 shard의 두 가지 유형이 있습니다. 문서가 인덱스로 인덱싱되면 기본 shard 중 하나에 저장되고 해당 shard의 모든 복제본에 복사됩니다. 기본 shard의 수는 인덱스가 생성될 때 지정되며 인덱스 수명 중에는 변경할 수 없습니다. 언제든지 복제본 shard 수를 변경할 수 있습니다.

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

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

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

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

표 7.3. Elasticsearch 클러스터 shard 상태 차트

지표	설명
클러스터 활성 shard	클러스터의 활성 기본 shard 수 및 복제본을 포함한 총 shard 수입니다. shard 수가 증가하면 클러스터 성능이 저하되기 시작할 수 있습니다.
클러스터 초기화 shard	클러스터의 비활성 shard 수입니다. 비활성 shard는 초기화 중이거나 다른 노드에 재 할당되거나 할당되지 않은 shard입니다. 일반적으로 클러스터에는 짧은 기간 동안 비활성 shard가 있습니다. 장기간에 걸쳐 비활성 shard 수가 증가하면 문제를 나타낼 수 있습니다.
클러스터 재배치 shard	Elasticsearch가 새 노드로 재배치하는 shard 수입니다. Elasticsearch는 노드의 메모리 사용량이 많거나 클러스터에 새 노드를 추가한 경우 등 여러 가지 이유로 노드를 재배치합니다.

지표	설명
할당되지 않은 shard 를 러스터	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 배치합니다.

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

표 7.4. 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.
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.

지표	설명
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

Fielddata는 인덱스의 용어 목록을 보유하고 **JVM** 힙에 보관되는 **Elasticsearch** 데이터 구조입니다. 필드 데이터 구축은 비용이 많이 드는 작업이므로 **Elasticsearch**는 필드 데이터 구조를 캐시합니다. **Elasticsearch**는 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 7.5. 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 **Elasticsearch**에서 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 7.6. Elasticsearch 노드 쿼리 차트

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

Elasticsearch 인덱스 제한

문서를 인덱싱할 때 **Elasticsearch**는 데이터의 물리적 표현인 인덱스 세그먼트에 문서를 저장합니다. 동시에 **Elasticsearch**는 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 **Elasticsearch**는 일반적으로 인덱싱에 할당된 스레드 수를 단일 스레드로 줄여 인덱싱을 제한합니다.

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

표 7.7. 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 힘 통계

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

표 7.8. JVM 힘 통계 차트

지표	설명

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

7.4. KIBANA를 사용한 로그 시각화

ElasticSearch 로그 저장소를 사용하는 경우 **Kibana** 콘솔을 사용하여 수집된 로그 데이터를 시각화할 수 있습니다.

Kibana를 사용하면 데이터로 다음을 수행할 수 있습니다.

- 검색 템을 사용하여 데이터를 검색하고 찾습니다.
- 시각화 템을 사용하여 데이터를 차트로 작성하고 매핑 합니다.
- 대시보드 템을 사용하여 사용자 정의 대시보드를 생성하고 봅니다.

Kibana 인터페이스의 사용 및 구성은 이 문서의 범위를 벗어납니다. 인터페이스 사용에 대한 자세한 내용은 [Kibana 설명서](#)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 기본 출력을 사용하는 파이프라인을 구성해야 합니다.

7.4.1. Kibana 인덱스 패턴 정의

인덱스 패턴은 시각화하려는 **Elasticsearch** 인덱스를 정의합니다. **Kibana**에서 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 **Pod**와 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 --subresource log -n <project>
```

출력 예

yes

참고

감사 로그는 기본적으로 내부 **OpenShift Container Platform Elasticsearch** 인스턴스에 저장되지 않습니다. **Kibana**에서 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.

절차

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 **Kibana**에 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 **Kibana**에 로그인 할 때 인덱스 패턴을 생성해야 합니다.

3.

새로운 인덱스 패턴에서 **Kibana** 시작화를 생성합니다.

7.4.2. Kibana에서 클러스터 로그 보기

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

사전 요구 사항

- **Red Hat OpenShift Logging** 및 **Elasticsearch Operator**가 설치되어 있어야 합니다.
- **Kibana** 인덱스 패턴이 있어야 합니다.
- **Kibana**에서 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 --subresource 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 템을 클릭하여 해당 문서에 대한 로그 항목을 표시합니다.

예 7.1. 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": "YmJmYTBINDkZTRmLTlIMGQtMjE3NmFiOGUyOWM3",
    "openshift": {
      "pod_ip": "10.0.182.28"
    }
  }
}
```

```

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

```

7.4.3. Kibana 구성

ClusterLogging 사용자 정의 리소스(CR)를 수정하여 Kibana 콘솔을 사용하여 구성할 수 있습니다.

7.4.3.1. CPU 및 메모리 제한 구성

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

프로세스

1.

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

```
$ oc -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.

7.4.3.2. 로그 시각화 프로그램 노드의 확장성 증복

증복성에 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

Kibana 노드의 수를 지정합니다.

8장. 로깅 배포 구성

8.1. 로깅 구성 요소에 대한 CPU 및 메모리 제한 구성

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

8.1.1. CPU 및 메모리 제한 구성

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

프로세스

1.

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

```
$ oc -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.

8.2. 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.

8.2.1. 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 Logging**이 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 설정과 함께 `/etc/systemd/journald.conf` 파일을 포함하는 **Butane** 구성 파일 **40-worker-custom-journald.bu**를 만듭니다.



참고

Butane에 대한 자세한 내용은 “**Butane**을 사용하여 머신 구성 생성”을 참조하십시오.

```
variant: openshift
version: 4.12.0
metadata:
  name: 40-worker-custom-journald
  labels:
    machineconfiguration.openshift.io/role: "worker"
storage:
  files:
    - path: /etc/systemd/journald.conf
      mode: 0644 ①
      overwrite: true
      contents:
        inline: |
          Compress=yes ②
          ForwardToConsole=no ③
          ForwardToSyslog=no
          MaxRetentionSec=1month ④
          RateLimitBurst=10000 ⑤
          RateLimitIntervalSec=30s
          Storage=persistent ⑥
          SyncIntervalSec=1s ⑦
          SystemMaxUse=8G ⑧
          SystemKeepFree=20% ⑨
          SystemMaxFileSize=10M ⑩
```

1

journald.conf 파일에 대한 권한을 설정합니다. **0644** 권한을 설정하는 것이 좋습니다.

2

로그를 파일 시스템에 쓰기 전에 압축할지 여부를 지정합니다. 메시지를 압축하려면 **yes**를 지정하고 압축하지 않으려면 **no**를 지정합니다. 기본값은 **yes**입니다.

3

로그 메시지를 전달할지 여부를 구성합니다. 각각에 대해 기본값은 **no**입니다. 다음을 지정합니다.

- 시스템 콘솔에 로그를 전달하려면 **ForwardToConsole**을 지정합니다.
- 로그를 커널 로그 버퍼로 전달하려면 **ForwardToKMsg**를 지정합니다.
- **syslog** 데몬으로 전달하려면 **ForwardToSyslog**를 지정합니다.
- 로그인한 모든 사용자에게 월(**wall**) 메시지로 메시지를 전달하려면 **ForwardToWall**을 지정합니다.

4

저널 항목을 저장할 최대 시간을 지정합니다. 초를 지정하려면 숫자를 입력합니다. 또는 "year", "month", "week", "day", "h" 또는 "m"과 같은 단위를 포함합니다. 비활성화하려면 **0**을 입력합니다. 기본값은 **1month**입니다.

5

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.

6

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

- `/run/log/journal/`의 메모리에 로그를 저장하는 **volatile**입니다. 이러한 로그는 재부팅 후 손실됩니다.
- `/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.

7

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.

8

저널이 사용할 수 있는 최대 크기를 지정합니다. 기본값은 **8G**입니다.

9

시스템에서 사용 가능한 디스크 공간을 지정합니다. 기본값은 **20%**입니다.

10

`/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.

오. 해당 페이지에 나열된 기본 설정은 OpenShift Container Platform에 적용되지 않을 수 있습니다.

2.

Butane을 사용하여 노드로 전달할 구성이 포함된 MachineConfig 개체 파일 40-worker-custom-journald.yaml을 생성합니다.

```
$ butane 40-worker-custom-journald.bu -o 40-worker-custom-journald.yaml
```

3.

머신 구성을 적용합니다. 예를 들면 다음과 같습니다.

```
$ oc apply -f 40-worker-custom-journald.yaml
```

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

4.

각 노드에 새로 렌더링된 구성의 롤아웃 상태를 모니터링합니다.

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

출력 예

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

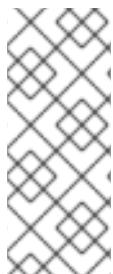
...

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

9장. 로그 수집 및 전달

9.1. 로그 수집 및 전달 정보

Red Hat OpenShift Logging Operator는 **ClusterLogForwarder** 리소스 사양에 따라 수집기를 배포합니다. 이 Operator에서 지원하는 수집기 옵션은 레거시 **Fluentd** 수집기와 벡터 수집기의 두 가지입니다.



참고

Fluentd는 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다. **Red Hat**은 현재 릴리스 라이프사이클 동안 이 기능에 대한 버그 수정 및 지원을 제공하지만 이 기능은 더 이상 개선 사항을 받지 않습니다. **Fluentd** 대신 **Vector**를 사용할 수 있습니다.

9.1.1. 로그 컬렉션

로그 수집기는 각 **OpenShift Container Platform** 노드에 **Pod**를 배포하여 컨테이너 및 노드 로그를 수집하는 데몬 세트입니다.

기본적으로 로그 수집기는 다음 소스를 사용합니다.

- 운영 체제, 컨테이너 런타임 및 **OpenShift Container Platform**의 **journald** 로그 메시지에 의해 생성된 시스템 및 인프라 로그입니다.
- 모든 컨테이너 로그에 대한 **/var/log/containers/*.log**

감사 로그를 수집하도록 로그 수집기를 구성하는 경우 **/var/log/audit/audit.log**에서 해당 로그를 수집합니다.

로그 수집기는 이러한 소스에서 로그를 수집하여 로깅 구성에 따라 내부 또는 외부로 전달합니다.

9.1.1.1. 로그 수집기 유형

베타 는 로깅을 위해 **Fluentd**의 대안으로 제공되는 로그 수집기입니다.

ClusterLogging 사용자 정의 리소스(**CR**) 컬렉션 사양을 수정하여 클러스터에서 사용하는 로깅 수집기 유형을 구성할 수 있습니다.

Vector를 수집기로 구성하는 **ClusterLogging CR**의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
  name: instance
  namespace: openshift-logging
spec:
  collection:
    logs:
      type: vector
      vector: {}
# ...
```

9.1.1.2. 로그 컬렉션 제한 사항

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 **Pod**를 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 간주됨을 의미합니다.



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.

9.1.1.3. 유형별 로그 수집기 기능

표 9.1. 로그 소스

기능	fluentd	vector
앱 컨테이너 로그	✓	✓
앱별 라우팅	✓	✓
네임스페이스별 앱별 라우팅	✓	✓
인프라 컨테이너 로그	✓	✓
인프라 저널 로그	✓	✓
kube API 감사 로그	✓	✓
OpenShift API 감사 로그	✓	✓
OVN(Open Virtual Network) 감사 로그	✓	✓

표 9.2. 권한 부여 및 인증

기능	fluentd	vector
Elasticsearch 인증서	✓	✓
Elasticsearch 사용자 이름 / 암호	✓	✓
NetNamespace 키	✓	✓
<.> STS	✓	✓
Kafka 인증서	✓	✓
Kafka 사용자 이름 / 암호	✓	✓
kafka SASL	✓	✓
Loki bearer 토큰	✓	✓

표 9.3. Normalizations 및 transformationss

기능	fluentd	vector
Viaq 데이터 모델 - 앱	✓	✓
Viaq 데이터 모델 - 인프라	✓	✓

기능	fluentd	vector
Viaq 데이터 모델 - 인프라(journal)	✓	✓
Viaq 데이터 모델 - Linux 감사	✓	✓
Viaq 데이터 모델 - kube-apiserver audit	✓	✓
Viaq 데이터 모델 - OpenShift API 감사	✓	✓
Viaq 데이터 모델 - OVN	✓	✓
loglevel Normalization	✓	✓
JSON 구문 분석	✓	✓
구조화된 인덱스	✓	✓
다중 줄 오류 감지	✓	✓
multicontainer / split 인덱스	✓	✓
flatten 라벨	✓	✓
CLF 정적 레이블	✓	✓

표 9.4. tuning

기능	fluentd	vector
fluentd readlinelimit	✓	
fluentd 버퍼	✓	
- chunklimitsize	✓	
- 총 제한 크기	✓	
- overflowaction	✓	
- flushthreadcount	✓	
- flushmode	✓	
- flushinterval	✓	

기능	fluentd	vector
----	---------	--------

- retrywait	✓	
- retrytype	✓	
- retrymaxinterval	✓	
- retrytimeout	✓	

표 9.5. 가시성

기능	fluentd	vector
지표	✓	✓
대시보드	✓	✓
경고	✓	✓

표 9.6. 기타

기능	fluentd	vector
글로벌 프록시 지원	✓	✓
x86 지원	✓	✓
ARM 지원	✓	✓
IBM Power 지원	✓	✓
IBM Z 지원	✓	✓
IPv6 지원	✓	✓
로그 이벤트 버퍼링	✓	
연결이 끊긴 클러스터	✓	✓

9.1.1.4. 수집기 출력

다음 컬렉터 출력이 지원됩니다.

표 9.7. 지원되는 출력

기능	fluentd	vector
Elasticsearch v6-v8	✓	✓
fluent forward	✓	
Syslog RFC3164	✓	ECDHE (logging 5.7 이상)
Syslog RFC5424	✓	ECDHE (logging 5.7 이상)
kafka	✓	✓
서드니	✓	✓
<.> STS	✓	✓
Loki	✓	✓
HTTP	✓	ECDHE (logging 5.7 이상)
Google 클라우드 로깅	✓	✓
Splunk		Cryostat (logging 5.6 이상)

9.1.2. 로그 전송

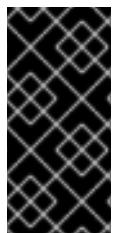
관리자는 수집되는 로그, 변환 방법, 전달되는 위치를 지정하는 **ClusterLogForwarder** 리소스를 생성할 수 있습니다.

ClusterLogForwarder 리소스는 컨테이너, 인프라 및 감사 로그를 클러스터 내부 또는 외부의 특정 끝점으로 전달하는 데 사용할 수 있습니다. **TLS(Transport Layer Security)**가 지원되므로 로그를 안전하게 전송하도록 로그 전달자를 구성할 수 있습니다.

관리자는 어떤 유형의 로그에 액세스하고 전달할 수 있는 서비스 계정 및 사용자를 정의하는 **RBAC** 권한을 부여할 수도 있습니다.

9.1.2.1. 로그 전달 구현

레거시 구현과 다중 로그 전달 기능이라는 두 가지 로그 전달 구현을 사용할 수 있습니다.



중요

다중 로그 전달자 기능과 함께 사용할 수 있도록 **Vector** 수집기만 지원됩니다.
Fluentd 수집기는 레거시 구현에서만 사용할 수 있습니다.

9.1.2.1.1. 기존 구현

기존 구현에서는 클러스터에서 하나의 로그 전달자만 사용할 수 있습니다. 이 모드의 **ClusterLogForwarder** 리소스는 **instance** 여야 하며 **openshift-logging** 네임스페이스에서 생성해야 합니다. **ClusterLogForwarder** 리소스에는 **openshift-logging** 네임스페이스에 **instance**라는 해당 **ClusterLogging** 리소스도 필요합니다.

9.1.2.1.2. 다중 로그 전달자 기능

다중 로그 전달자 기능은 로깅 5.8 이상에서 사용할 수 있으며 다음과 같은 기능을 제공합니다.

- 관리자는 로그 컬렉션을 정의할 수 있는 사용자와 수집할 수 있는 로그를 제어할 수 있습니다.
- 필요한 권한이 있는 사용자는 추가 로그 컬렉션 구성을 지정할 수 있습니다.
- 더 이상 사용되지 않는 **Fluentd** 수집기에서 **Vector** 수집기로 마이그레이션하는 관리자는 새 로그 전달자를 기존 배포와 별도로 배포할 수 있습니다. 워크로드가 마이그레이션되는 동안 기존 및 새 로그 전달자가 동시에 작동할 수 있습니다.

다중 로그 전달자 구현에서는 **ClusterLogForwarder** 리소스에 대한 해당 **ClusterLogging** 리소스를 생성할 필요가 없습니다. 다음 예외를 제외하고 모든 네임스페이스에서 모든 이름을 사용하여 여러 **ClusterLogForwarder** 리소스를 생성할 수 있습니다.

- **Fluentd** 수집기를 사용하는 레거시 워크플로우를 지원하는 로그 전달자를 위해 예약되어 있으므로 **openshift-logging** 네임스페이스에 **instance**라는 **ClusterLogForwarder** 리소스를 생성할 수 없습니다.

- 수집기용으로 예약되어 있으므로 **openshift-logging** 네임스페이스에서 **collector**라는 **ClusterLogForwarder** 리소스를 생성할 수 없습니다.

9.1.2.2. 클러스터의 다중 로그 전달자 기능 활성화

다중 로그 전달자 기능을 사용하려면 해당 서비스 계정에 대한 서비스 계정 및 클러스터 역할 바인딩을 생성해야 합니다. 그런 다음 **ClusterLogForwarder** 리소스에서 서비스 계정을 참조하여 액세스 권한을 제어할 수 있습니다.



중요

openshift-logging 네임스페이스 이외의 추가 네임스페이스에서 다중 로그 전달을 지원하려면 모든 네임스페이스를 조사하도록 **Red Hat OpenShift Logging Operator**를 업데이트해야 합니다. 이 기능은 새로운 **Red Hat OpenShift Logging Operator** 버전 5.8 설치에서 기본적으로 지원됩니다.

9.1.2.2.1. 로그 컬렉션 RBAC 권한 승인

로깅 5.8 이상에서 **Red Hat OpenShift Logging Operator**는 **collect-audit-logs**, **collect-application-logs**, **collect-infrastructure-logs** 클러스터 역할을 제공하여 수집기가 감사 로그, 애플리케이션 로그 및 인프라 로그를 각각 수집할 수 있습니다.

필요한 클러스터 역할을 서비스 계정에 바인딩하여 로그 컬렉션에 대한 **RBAC** 권한을 부여할 수 있습니다.

사전 요구 사항

- Red Hat OpenShift Logging Operator**는 **openshift-logging** 네임스페이스에 설치됩니다.
- 관리자 권한이 있습니다.

절차

- 수집기의 서비스 계정을 생성합니다. 인증을 위해 토큰이 필요한 스토리지에 로그를 작성하면서 서비스 계정에 토큰을 포함해야 합니다.

2.

적절한 클러스터 역할을 서비스 계정에 바인딩합니다.

바인딩 명령 예

```
$ oc adm policy add-cluster-role-to-user <cluster_role_name> system:serviceaccount:<namespace_name>:<service_account_name>
```

추가 리소스

- [RBAC를 사용하여 권한 정의 및 적용](#)
- [애플리케이션에서 서비스 계정 사용](#)
- [RBAC 권한 부여 Kubernetes 문서 사용](#)

9.2. 로그 출력 유형

출력은 로그 전달자에서 로그를 전송하는 대상을 정의합니다. **ClusterLogForwarder CR**(사용자 정의 리소스)에서 여러 유형의 출력을 구성하여 다른 프로토콜을 지원하는 서버로 로그를 보낼 수 있습니다.

9.2.1. 지원되는 로그 전달 출력

출력은 다음 유형 중 하나일 수 있습니다.

표 9.8. 지원되는 로그 출력 유형

출력 유형	프로토콜	테스트에 사용	로깅 버전	지원되는 수집기 유형
Elasticsearch v6	HTTP 1.1	6.8.1, 6.8.23	5.6+	Fluentd, Vector
Elasticsearch v7	HTTP 1.1	7.12.2, 7.17.7, 7.10.1	5.6+	Fluentd, Vector

출력 유형	프로토콜	테스트에 사용	로깅 버전	지원되는 수집기 유형
Elasticsearch v8	HTTP 1.1	8.4.3, 8.6.1	5.6+	fluentd [1], 벡터
fluent Forward	Fluentd 전달 v1	Fluentd 1.14.6, Logstash 7.10.1, Fluentd 1.14.5	5.4+	fluentd
Google 클라우드 로깅	HTTPS를 통한 REST	latest	5.7+	vector
HTTP	HTTP 1.1	Fluentd 1.14.6, 벡터 0.21	5.7+	Fluentd, Vector
kafka	Kafka 0.11	Kafka 2.4.1, 2.7.0, 3.3.1	5.4+	Fluentd, Vector
Loki	HTTP 및 HTTPS를 통한 REST	2.3.0, 2.5.0, 2.7, 2.2.1	5.4+	Fluentd, Vector
Splunk	HEC	8.2.9, 9.0.0	5.7+	vector
syslog	RFC3164, RFC5424	Rsyslog 8.37.0-9.el7, rsyslog-8.39.0	5.4+	Fluentd, Vector [2]
Amazon CloudWatch	HTTPS를 통한 REST	latest	5.4+	Fluentd, Vector

1.

Fluentd는 로깅 버전 5.6.2에서 Elasticsearch 8을 지원하지 않습니다.

2.

벡터는 로깅 버전 5.7 이상에서 Syslog를 지원합니다.

9.2.2. 출력 유형 설명

default

클러스터 내 Red Hat 관리 로그 저장소입니다. 기본 출력을 구성할 필요는 없습니다.



참고

기본 출력 이름이 클러스터 on-cluster인 Red Hat 관리 로그 저장소를 참조하도록 예약되어 있으므로 기본 출력을 구성하는 경우 오류 메시지가 표시됩니다.

loki

수평으로 확장 가능한 고가용성 다중 테넌트 로그 집계 시스템인 **Loki**입니다.

kafka

Kafka 브로커. **kafka** 출력은 **TCP** 또는 **TLS** 연결을 사용할 수 있습니다.

elasticsearch

외부 **Elasticsearch** 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 **TLS**를 포함하거나 포함하지 않고 사용할 수 있습니다.



중요

fluentdForward 출력은 **Fluentd** 수집기를 사용하는 경우에만 지원됩니다. **Vector** 수집기를 사용하는 경우에는 지원되지 않습니다. **Vector** 수집기를 사용하는 경우 **http** 출력을 사용하여 로그를 **Fluentd**로 전달할 수 있습니다.

syslog

syslog RFC3164 또는 **RFC5424** 프로토콜을 지원하는 외부 로그 집계 솔루션입니다. **syslog** 출력은 **UDP**, **TCP** 또는 **TLS** 연결을 사용할 수 있습니다.

cloudwatch

AWS(Amazon Web Services)에서 호스팅하는 모니터링 및 로그 스토리지 서비스인 **Amazon CloudWatch**입니다.

9.3. JSON 로그 전달 활성화

JSON 문자열을 구조화된 오브젝트로 구문 분석하도록 **Log Forwarding API**를 구성할 수 있습니다.

9.3.1. JSON 로그 구문 분석

ClusterLogForwarder 오브젝트를 사용하여 JSON 로그를 구조화된 오브젝트로 구문 분석하고 지원되는 출력으로 전달할 수 있습니다.

이 작동 방식을 설명하기 위해 다음과 같은 구조화된 JSON 로그 항목이 있다고 가정합니다.

구조화된 JSON 로그 항목 예

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

JSON 로그를 구문 분석할 수 있도록 다음 예와 같이 **parse: json** 을 **ClusterLogForwarder CR**의 파이프라인에 추가합니다.

parse: json을 보여주는 코드 조각 예

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

parse: json 을 사용하여 JSON 로그를 구문 분석할 때 CR은 다음 예와 같이 **structured** 필드에 JSON 구조화된 로그 항목을 복사합니다.

구조화된 JSON 로그 항목이 포함된 structured 출력 예

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



중요

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

9.3.2. Elasticsearch의 JSON 로그 데이터 구성

JSON 로그가 두 개 이상의 스키마를 따르는 경우 단일 인덱스에 저장하면 유형 충돌 및 카디널리티 문제 가 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** 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 CR**의 **pipeline.label.<key>** 요소이며 인덱스 이름을 구성하는 데 사용되는 값이 있습니다.

- **kubernetes.container_name**은 컨테이너 이름을 사용하여 인덱스 이름을 구성합니다.
- **structuredTypeName: structuredTypeKey** 필드가 설정되지 않았거나 키가 없으면 **structured** 유형으로 **structuredTypeName** 값이 사용됩니다. **structuredTypeKey** 필드와 **structuredTypeName** 필드를 함께 사용하면 **structuredTypeKey** 필드의 키가 JSON 로그 데이터에서 누락된 경우 **structuredTypeName** 값은 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.

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
# ...
spec:
# ...
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 사용하는것이 아니라 별도의 로그 형식에만 사용합니다. 예를 들어 대부분의 **Apache** 애플리케이션은 **LogApache**와 같은 동일한 **JSON** 로그 형식과 구조화된 유형을 사용합니다.

9.3.3. 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 그룹화 합니다. 이렇게 하면 **Elasticsearch**는 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.

structuredTypeKey 필드를 사용하여 로그 레코드 필드 중 하나를 지정합니다.

3.

structuredTypeName 필드를 사용하여 이름을 지정합니다.



중요

JSON 로그를 구문 분석하려면 **structuredTypeKey** 및 **structuredTypeName** 필드를 모두 설정해야 합니다.

4.

inputRefs의 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 <filename>.yaml
```

Red Hat OpenShift Logging Operator는 수집기 Pod를 재배포합니다. 그러나 재배포되지 않으면 수집기 Pod를 삭제하여 강제로 재배포합니다.

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

9.3.4. 동일한 Pod의 컨테이너에서 JSON 로그를 전달하여 인덱스를 분리

동일한 Pod 내의 다른 컨테이너에서 다른 인덱스로 구조화된 로그를 전달할 수 있습니다. 이 기능을 사용하려면 다중 컨테이너 지원을 사용하여 파이프라인을 구성하고 Pod에 주석을 달아야 합니다. 로그는 접두사가 **app-** 인 인덱스에 작성됩니다. Elasticsearch는 이를 수용할 수 있도록 별칭으로 구성하는 것이 좋습니다.



중요

JSON 형식의 로그는 애플리케이션에 따라 다릅니다. 너무 많은 인덱스를 생성하면 성능에 영향을 미치기 때문에 이 기능을 사용하여 호환되지 않는 JSON 형식의 로그 인덱스를 생성할 수 있습니다. 쿼리를 사용하여 서로 다른 네임스페이스 또는 호환되는 JSON 형식의 애플리케이션을 분리합니다.

사전 요구 사항

-

Red Hat OpenShift용 로깅: 5.5

절차

1.

ClusterLogForwarder CR 오브젝트를 정의하는 YAML 파일을 생성하거나 편집합니다.

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: instance
  namespace: openshift-logging
spec:
  outputDefaults:
    elasticsearch:
      structuredTypeKey: kubernetes.labels.logFormat ①
```

```

structuredTypeName: nologformat
enableStructuredContainerLogs: true ②
pipelines:
- inputRefs:
  - application
  name: application-logs
  outputRefs:
  - default
  parse: json

```

1

Kubernetes logFormat 레이블로 구성된 키-값 쌍의 값을 사용합니다.

2

다중 컨테이너 출력을 활성화합니다.

2.

Pod CR 오브젝트를 정의하는 YAML 파일을 생성하거나 편집합니다.

```

apiVersion: v1
kind: Pod
metadata:
  annotations:
    containerType.logging.openshift.io/heavy: heavy ①
    containerType.logging.openshift.io/low: low
spec:
  containers:
  - name: heavy ②
    image: heavyimage
  - name: low
    image: lowimage

```

1

Format: containerType.logging.openshift.io/<container-name>: <index>

2

주석 이름은 컨테이너 이름과 일치해야 합니다.



주의

이 구성은 클러스터의 **shard** 수를 크게 늘릴 수 있습니다.

추가 리소스

- [Kubernetes 주석](#)

추가 리소스

- [로그 전송 정보](#)

9.4. 로그 전달 구성

로깅 배포에서 컨테이너 및 인프라 로그는 기본적으로 **ClusterLogging** 사용자 정의 리소스(**CR**)에 정의된 내부 로그 저장소로 전달됩니다.

감사 로그는 보안 스토리지를 제공하지 않기 때문에 기본적으로 내부 로그 저장소로 전달되지 않습니다. 감사 로그를 전달하는 시스템이 조직 및 정부 규정을 준수하고 올바르게 보호되도록 할 책임이 있습니다.

이 기본 구성이 요구 사항을 충족하는 경우 **ClusterLogForwarder CR**을 구성할 필요가 없습니다. **ClusterLogForwarder CR**이 있는 경우 기본 출력이 포함된 파이프라인을 정의하지 않는 한 로그는 내부 로그 저장소로 전달되지 않습니다.

9.4.1. 타사 시스템으로 로그 전달 정보

OpenShift Container Platform 클러스터 내부 및 외부의 특정 끝점에 로그를 보내려면 **ClusterLogForwarder CR**(사용자 정의 리소스)에서 출력 및 파이프라인 조합을 지정합니다. 입력을 사용하여 특정 프로젝트와 관련된 애플리케이션 로그를 끝점으로 전달할 수도 있습니다. 인증은 **Kubernetes Secret** 오브젝트에서 제공합니다.

pipeline

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.** 노드 감사 시스템, `auditd`, Kubernetes API 서버, OpenShift API 서버 및 OVN 네트워크에서 생성된 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.

input

특정 프로젝트와 관련된 애플리케이션 로그를 파이프라인으로 전달합니다.

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

Secret

사용자 자격 증명과 같은 기밀 데이터를 포함하는 **key:value** 맵입니다.

다음을 확인합니다.

- 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 프로

토콜을 지원하는 서버에 로그를 보낼 수 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: <log_forwarder_name> ①
  namespace: <log_forwarder_namespace> ②
spec:
  serviceAccountName: <service_account_name> ③
  outputs:
    - name: elasticsearch-secure ④
      type: "elasticsearch"
      url: https://elasticsearch.secure.com:9200
      secret:
        name: elasticsearch
    - name: elasticsearch-insecure ⑤
      type: "elasticsearch"
      url: http://elasticsearch.insecure.com:9200
    - name: kafka-app ⑥
      type: "kafka"
      url: tls://kafka.secure.com:9093/app-topic
  inputs: ⑦
    - name: my-app-logs
      application:
        namespaces:
          - my-project
  pipelines:
    - name: audit-logs ⑧
      inputRefs:
        - audit
      outputRefs:
        - elasticsearch-secure
        - default
      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

레거시 구현에서 CR 이름은 인스턴스 여야 합니다. 다중 로그 전달자 구현에서는 모든 이름을 사용할 수 있습니다.

2

레거시 구현에서 CR 네임스페이스는 **openshift-logging** 이어야 합니다. 다중 로그 전달자 구현에서는 모든 네임스페이스를 사용할 수 있습니다.

3

서비스 계정의 이름입니다. 서비스 계정은 로그 전달자가 **openshift-logging** 네임스페이스에 배포되지 않은 경우에만 다중 로그 전달자 구현에 필요합니다.

4

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.

5

안전하지 않은 Elasticsearch 출력에 대한 구성:

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

6

보안 URL을 통한 클라이언트 인증 TLS 통신을 사용한 Kafka 출력 구성:

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

7

my-project 네임스페이스에서 애플리케이션 로그를 필터링하기 위한 입력 구성입니다.

8

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

- 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.

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.

- 파이프라인을 설명하는 이름입니다.
- **inputRefs**는 특정 입력인 **my-app-logs**입니다.
- **outputRefs**는 **default**입니다.
- 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.

지원되는 권한 부여 키

공통 키 유형이 여기에 제공됩니다. 일부 출력 유형은 출력별 구성 필드에 설명된 추가 특수 키를 지원합니다. 모든 보안 키는 선택 사항입니다. 관련 키를 설정하여 원하는 보안 기능을 활성화합니다. 키 및 시크릿, 서비스 계정, 포트 열기 또는 전역 프록시 구성과 같이 외부 대상에 필요할 수 있는 추가 구성을 생성하고 유지보수할 책임이 있습니다. **Open Shift Logging**은 권한 부여 조합이 일치하지 않는지 확인하지 않습니다.

TLS(Transport Layer Security)

시크릿 없이 **TLS URL**(**http://...** 또는 **ssl://...**)을 사용하면 기본 **TLS** 서버 측 인증이 활성화됩니다. 시크릿을 포함하고 다음 선택적 필드를 설정하여 추가 **TLS** 기능을 활성화합니다.

- **passphrase:** (문자열)로 인코딩된 **TLS** 개인 키를 디코딩합니다. **tls.key** 가 필요합니다.
- **ca-bundle.crt:** (문자열) 서버 인증을 위한 고객 **CA**의 파일 이름입니다.

사용자 이름 및 암호

- 사용자 이름:(문자열) 인증 사용자 이름입니다. 암호 가 필요합니다.
- **password:** (문자열) 인증 암호입니다. 사용자 이름이 필요합니다.

Simple Authentication Security Layer (SASL)

- **SASL.enable (boolean)**은 SASL을 활성화 또는 비활성화합니다. 누락된 경우 다른 **sasl.** 키가 설정되면 SASL이 자동으로 활성화됩니다.
- **SASL.mechanisms:** 허용 된 SASL 메커니즘 이름의 목록. 누락되거나 비어 있는 경우 시스템 기본값이 사용됩니다.
- **SASL.allow-insecure: (boolean)** 일반 텍스트 암호를 보내는 메커니즘을 허용합니다. 기본값은 **false**입니다.

9.4.1.1. 보안 생성

다음 명령을 사용하여 인증서 및 키 파일이 포함된 디렉터리에 보안을 생성할 수 있습니다.

```
$ oc create secret generic -n <namespace> <secret_name> \
--from-file=ca-bundle.crt=<your_bundle_file> \
--from-literal=username=<your_username> \
--from-literal=password=<your_password>
```



참고

최상의 결과를 위해 일반 또는 불투명 보안이 권장됩니다.

9.4.2. 로그 전달자 생성

로그 전달자를 생성하려면 서비스 계정에서 수집할 수 있는 로그 입력 유형을 지정하는 **ClusterLogForwarder CR**을 생성해야 합니다. 로그를 전달할 수 있는 출력을 지정할 수도 있습니다. 다중 로그 전달자 기능을 사용하는 경우 **ClusterLogForwarder CR**의 서비스 계정도 참조해야 합니다.

클러스터에서 다중 로그 전달자 기능을 사용하는 경우 이름을 사용하여 모든 네임스페이스에서 **ClusterLogForwarder** 사용자 정의 리소스(CR)를 생성할 수 있습니다. 레거시 구현을 사용하는 경우 **ClusterLogForwarder CR**의 이름은 **instance** 여야 하며 **openshift-logging** 네임스페이스에서 생성해야 합니다.



중요

ClusterLogForwarder CR을 생성하는 네임스페이스에 대한 관리자 권한이 필요합니다.

ClusterLogForwarder 리소스 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: <log_forwarder_name> ①
  namespace: <log_forwarder_namespace> ②
spec:
  serviceAccountName: <service_account_name> ③
  pipelines:
    - inputRefs:
        - <log_type> ④
      outputRefs:
        - <output_name> ⑤
    outputs:
      - name: <output_name> ⑥
        type: <output_type> ⑦
        url: <log_output_url> ⑧
# ...
```

1

레거시 구현에서 CR 이름은 인스턴스 여야 합니다. 다중 로그 전달자 구현에서는 모든 이름을 사용할 수 있습니다.

2

레거시 구현에서 CR 네임스페이스는 **openshift-logging** 이어야 합니다. 다중 로그 전달자 구현에서는 모든 네임스페이스를 사용할 수 있습니다.

3

서비스 계정의 이름입니다. 서비스 계정은 로그 전달자가 **openshift-logging** 네임스페이스에 배포되지 않은 경우에만 다중 로그 전달자 구현에 필요합니다.

4

5 7

로그를 전달할 출력 유형입니다. 이 필드의 값은 기본 값, loki, kafka, elasticsearch, fluentdForward, syslog 또는 cloudwatch 일 수 있습니다.



참고

기본 출력 유형은 multi 로그 전달자 구현에서 지원되지 않습니다.

6

로그를 전달할 출력의 이름입니다.

8

로그를 전달할 출력의 URL입니다.

9.4.3. 다중 라인 예외 탐지 활성화

컨테이너 로그에 대한 여러 줄 오류 감지를 활성화합니다.



주의

이 기능을 활성화하면 성능에 영향을 미칠 수 있으며 추가 컴퓨팅 리소스 또는 대체 로깅 솔루션이 필요할 수 있습니다.

로그 구문 분석기에서는 별도의 예외와 동일한 예외의 별도의 행을 잘못 식별합니다. 이로 인해 추가 로그 항목이 생성되고 추적된 정보에 대한 불완전하거나 부정확한 보기가 발생합니다.

java 예외 예

```
java.lang.NullPointerException: Cannot invoke "String.toString()" because "<param1>" is null
at testjava.Main.handle(Main.java:47)
at testjava.Main.printMe(Main.java:19)
at testjava.Main.main(Main.java:10)
```

- 로깅을 사용하여 다중 줄 예외를 감지하고 단일 로그 항목으로 재조정할 수 있도록 하려면 **ClusterLogForwarder** 사용자 정의 리소스(CR)에 값이 **true** 인 **detectMultilineErrors** 필드가 포함되어 있는지 확인합니다.

ClusterLogForwarder CR의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: instance
  namespace: openshift-logging
spec:
  pipelines:
    - name: my-app-logs
      inputRefs:
        - application
      outputRefs:
        - default
      detectMultilineErrors: true
```

9.4.3.1. 세부 정보

로그 메시지가 예외 스택 추적을 형성하는 연속 시퀀스로 표시되면 단일 통합 로그 레코드로 결합됩니다. 첫 번째 로그 메시지의 콘텐츠는 순서에 있는 모든 메시지 필드의 연결 콘텐츠로 교체됩니다.

표 9.9. 수집기당 지원되는 언어:

언어	fluentd	vector
Java	✓	✓
JS	✓	✓

언어	fluentd	vector
Ruby	✓	✓
Python	✓	✓
Golang	✓	✓
PHP	✓	✓
Dart	✓	✓

9.4.3.2. 문제 해결

활성화되면 수집기 구성에 **type: detect_exceptions**가 있는 새 섹션이 포함됩니다.

벡터 구성 섹션의 예

```
[transforms.detect_exceptions_app-logs]
type = "detect_exceptions"
inputs = ["application"]
languages = ["All"]
group_by = ["kubernetes.namespace_name","kubernetes.pod_name","kubernetes.container_name"]
expire_after_ms = 2000
multiline_flush_interval_ms = 1000
```

fluentd config 섹션의 예

```
<label @MULTILINE_APP_LOGS>
<match kubernetes.>
  @type detect_exceptions
  remove_tag_prefix 'kubernetes'
  message message
  force_line_breaks true
  multiline_flush_interval .2
</match>
</label>
```

9.4.4. GCP(Google Cloud Platform)에 로그 전달

내부 기본 OpenShift Container Platform 로그 저장소에 추가하거나 대신 Google Cloud Logging에 로그를 전달할 수 있습니다.



참고

Fluentd에서 이 기능을 사용하는 것은 지원되지 않습니다.

사전 요구 사항

- Red Hat OpenShift Logging Operator 5.5.1 이상

절차

1.

[Google 서비스 계정 키](#)를 사용하여 시크릿을 생성합니다.

```
$ oc -n openshift-logging create secret generic gcp-secret --from-file google-application-credentials.json=<your_service_account_key_file.json>
```

2.

아래 템플릿을 사용하여 ClusterLogForwarder 사용자 정의 리소스 YAML을 생성합니다.

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: <log_forwarder_name> ①
  namespace: <log_forwarder_namespace> ②
spec:
  serviceAccountName: <service_account_name> ③
  outputs:
    - name: gcp-1
      type: googleCloudLogging
      secret:
        name: gcp-secret
      googleCloudLogging:
        projectId : "openshift-gce-devel" ④
        logId : "app-gcp" ⑤
  pipelines:
```

```

- name: test-app
  inputRefs: ⑥
    - application
  outputRefs:
    - gcp-1

```

1

레거시 구현에서 CR 이름은 인스턴스 여야 합니다. 다중 로그 전달자 구현에서는 모든 이름을 사용할 수 있습니다.

2

레거시 구현에서 CR 네임스페이스는 **openshift-logging** 이어야 합니다. 다중 로그 전달자 구현에서는 모든 네임스페이스를 사용할 수 있습니다.

3

서비스 계정의 이름입니다. 서비스 계정은 로그 전달자가 **openshift-logging** 네임스페이스에 배포되지 않은 경우에만 다중 로그 전달자 구현에 필요합니다.

4

GCP 리소스 계층에 로그를 저장하려는 위치에 따라 **projectId, folderId, organizationId** 또는 **billingAccountId** 필드 및 해당 값을 설정합니다.

5

Log Entry의 **logName** 필드에 추가할 값을 설정합니다.

6

파이프라인, 인프라 또는 감사 파이프라인을 사용하여 전달할 로그 유형을 지정합니다.

추가 리소스

- [Google 클라우드 청구 문서](#)
- [Google Cloud Logging 쿼리 언어 문서](#)

9.4.5. Splunk에 로그 전달

내부 기본 OpenShift Container Platform 로그 저장소에 추가하거나 대신 Splunk HTTP 이벤트 수집기(HEC)에 로그를 전달할 수 있습니다.



참고

Fluentd에서 이 기능을 사용하는 것은 지원되지 않습니다.

사전 요구 사항

- Red Hat OpenShift Logging Operator 5.6 이상
- 백터 가 수집기로 지정된 ClusterLogging 인스턴스
- base64로 인코딩된 Splunk HEC 토큰

절차

1.

Base64로 인코딩된 Splunk HEC 토큰을 사용하여 시크릿을 생성합니다.

```
$ oc -n openshift-logging create secret generic vector-splunk-secret --from-literal
hecToken=<HEC_Token>
```

2.

아래 템플릿을 사용하여 ClusterLogForwarder 사용자 정의 리소스(CR)를 생성하거나 편집합니다.

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: <log_forwarder_name> ①
  namespace: <log_forwarder_namespace> ②
spec:
  serviceAccountName: <service_account_name> ③
  outputs:
    - name: splunk-receiver ④
      secret:
        name: vector-splunk-secret ⑤
      type: splunk ⑥
      url: <http://your.splunk.hec.url:8088> ⑦
```

```

pipelines: ⑧
  - inputRefs:
    - application
    - infrastructure
  name: ⑨
  outputRefs:
    - splunk-receiver ⑩

```

1

레거시 구현에서 CR 이름은 인스턴스 여야 합니다. 다중 로그 전달자 구현에서는 모든 이름을 사용할 수 있습니다.

2

레거시 구현에서 CR 네임스페이스는 **openshift-logging** 이어야 합니다. 다중 로그 전달자 구현에서는 모든 네임스페이스를 사용할 수 있습니다.

3

서비스 계정의 이름입니다. 서비스 계정은 로그 전달자가 **openshift-logging** 네임스페이스에 배포되지 않은 경우에만 다중 로그 전달자 구현에 필요합니다.

4

출력 이름을 지정합니다.

5

HEC 토큰이 포함된 시크릿의 이름을 지정합니다.

6

출력 유형을 **splunk**로 지정합니다.

7

Splunk HEC의 URL(포트 포함)을 지정합니다.

8

파이프라인, 인프라 또는 감사 파이프라인을 사용하여 전달할 로그 유형을 지정합니다.

9

선택사항: 파이프라인의 이름을 지정합니다.

10

이 파이프라인으로 로그를 전달할 때 사용할 출력 이름을 지정합니다.

9.4.6. HTTP를 통해 로그 전달

HTTP를 통한 로그 전달은 **Fluentd** 및 **Vector** 로그 수집기 모두에서 지원됩니다. 활성화하려면 **ClusterLogForwarder CR**(사용자 정의 리소스)에서 **http**를 출력 유형으로 지정합니다.

절차

-

아래 템플릿을 사용하여 **ClusterLogForwarder CR**을 생성하거나 편집합니다.

ClusterLogForwarder CR의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: <log_forwarder_name> ①
  namespace: <log_forwarder_namespace> ②
spec:
  serviceAccountName: <service_account_name> ③
  outputs:
    - name: httpout-app
      type: http
      url: ④
      http:
        headers: ⑤
        h1: v1
        h2: v2
        method: POST
      secret:
        name: ⑥
      tls:
        insecureSkipVerify: ⑦
  pipelines:
    - name:
      inputRefs:
```

- application

outputRefs:

- 8

1

레거시 구현에서 CR 이름은 인스턴스 여야 합니다. 다중 로그 전달자 구현에서는 모든 이름을 사용할 수 있습니다.

2

레거시 구현에서 CR 네임스페이스는 **openshift-logging** 이어야 합니다. 다중 로그 전달자 구현에서는 모든 네임스페이스를 사용할 수 있습니다.

3

서비스 계정의 이름입니다. 서비스 계정은 로그 전달자가 **openshift-logging** 네임스페이스에 배포되지 않은 경우에만 다중 로그 전달자 구현에 필요합니다.

4

로그의 대상 주소입니다.

5

로그 레코드와 함께 전송할 추가 헤더입니다.

6

대상 자격 증명의 시크릿 이름입니다.

7

값은 **true** 또는 **false** 입니다.

8

이 값은 출력 이름과 동일해야 합니다.

9.4.7. 특정 프로젝트의 애플리케이션 로그 전달

내부 로그 저장소를 사용하거나 대신 특정 프로젝트의 애플리케이션 로그 사본을 외부 로그 수집기로 전달할 수 있습니다. **OpenShift Container Platform**에서 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 생성해야 합니다.

사전 요구 사항

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

절차

- ClusterLogForwarder CR**을 정의하는 YAML 파일을 생성하거나 편집합니다.

ClusterLogForwarder CR의 예

```
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 8
  pipelines:
    - name: forward-to-fluentd-insecure 9
      inputRefs: 10
```

```

- my-app-logs
outputRefs: ⑪
- fluentd-server-insecure
labels:
  project: "my-project" ⑫
- name: forward-to-fluentd-secure ⑬
inputRefs:
- application ⑭
- 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

출력의 이름입니다.

4

출력 유형: `elasticsearch`, `fluentdForward`, `syslog` 또는 `kafka`.

5

유효한 절대 URL인 외부 로그 수집기의 URL 및 포트입니다. 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 프로젝트의 애플리케이션 로그를 필터링하기 위한 입력 구성입니다.

8

네임스페이스를 지정하지 않으면 모든 네임스페이스에서 로그가 수집됩니다.

9

파이프라인 구성은 이름이 지정된 입력의 로그를 이름이 지정된 출력으로 보냅니다.
이 예에서 **forward-to-fluentd-insecure**라는 파이프라인은 **my-app-logs**라는 입력에서
fluentd-server-insecure라는 출력으로 로그를 전달합니다.

10

입력 목록입니다.

11

사용할 출력의 이름입니다.

12

선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

13

로그를 다른 로그 집계기로 보내는 파이프라인 구성입니다.

- 선택 사항: 파이프라인의 이름을 지정합니다.
- 파이프라인을 사용하여 전달할 로그 유형 (**application**, **infrastructure**, 또는 **audit**)을 지정합니다.
- 이 파이프라인으로 로그를 전달할 때 사용할 출력 이름을 지정합니다.
- 선택 사항: **default** 출력을 지정하여 로그를 기본 로그 저장소로 전달합니다.

-

선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

14

이 구성은 사용할 때 모든 네임스페이스의 애플리케이션 로그가 수집됩니다.

2.

다음 명령을 실행하여 **ClusterLogForwarder CR**을 적용합니다.

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

9.4.8. 특정 pod에서 애플리케이션 로그 전달

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

다양한 네임스페이스의 다른 **pod**와 함께 실행되는 **pod**로 구성된 애플리케이션이 있다고 가정합니다. 이러한 **pod**에 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 **Pod**를 모두 선택할 수 있어야 합니다.

절차

1.

ClusterLogForwarder CR 오브젝트를 정의하는 **YAML** 파일을 생성하거나 편집합니다. 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: <log_forwarder_name> ①
  namespace: <log_forwarder_namespace> ②
spec:
```

```

pipelines:
  - inputRefs: [ myAppLogData ] ③
    outputRefs: [ default ] ④
inputs: ⑤
  - name: myAppLogData
    application:
      selector:
        matchLabels: ⑥
        environment: production
        app: nginx
      namespaces: ⑦
        - app1
        - app2
outputs: ⑧
  - <output_name>
  ...

```

1

레거시 구현에서 CR 이름은 인스턴스 여야 합니다. 다중 로그 전달자 구현에서는 모든 이름을 사용할 수 있습니다.

2

레거시 구현에서 CR 네임스페이스는 **openshift-logging** 이어야 합니다. 다중 로그 전달자 구현에서는 모든 네임스페이스를 사용할 수 있습니다.

3

inputs[].name에서 하나 이상의 쉼표로 구분된 값을 지정합니다.

4

outputs[]에서 하나 이상의 쉼표로 구분된 값을 지정합니다.

5

고유한 pod 레이블 집합이 있는 각 애플리케이션에 대해 고유한 **inputs[].name**을 정의합니다.

6

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.

7

선택 사항: 하나 이상의 네임스페이스를 지정합니다.

8

로그 데이터를 전달할 출력을 하나 이상 지정합니다.

2.

선택 사항: 로그 데이터 수집을 특정 네임스페이스로 제한하려면 위 예제에 표시된 대로 `inputs[].name.application.namespaces`를 사용합니다.

3.

선택 사항: Pod 레이블이 다른 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 ]
```

4.

CR 오브젝트를 생성합니다.

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

추가 리소스

-

Kubernetes의 `matchLabels`에 대한 자세한 내용은 [세트 기반 요구 사항을 지원하는 리소스](#)를 참조하십시오.

9.4.9. API 감사 필터 개요

OpenShift API 서버는 각 API 호출에 대한 감사 이벤트를 생성하여 요청자의 요청, 응답 및 ID를 자세

히 설명하여 대량의 데이터를 생성합니다. **API** 감사 필터는 규칙을 사용하여 필수가 아닌 이벤트를 제외하고 이벤트 크기 감소를 활성화하여 보다 관리가 용이한 감사 추적을 지원합니다. 규칙이 순서대로 확인되어 첫 번째 일치 시 검사가 중지됩니다. 이벤트에 포함된 데이터 양은 수준 필드의 값에 따라 결정됩니다.

- 없음: 이벤트가 삭제되었습니다.
- **metadata:** 감사 메타데이터가 포함되어 요청 및 응답 본문이 제거됩니다.
- **요청:** 감사 메타데이터와 요청 본문이 포함되어 응답 본문이 제거됩니다.
- **RequestResponse:** 메타데이터, 요청 본문, 응답 본문 등 모든 데이터가 포함됩니다. 응답 본문은 매우 커질 수 있습니다. 예를 들어 `oc get pods -A` 는 클러스터의 모든 포드에 대한 YAML 설명이 포함된 응답 본문을 생성합니다.

로깅 5.8 이상에서 **ClusterLogForwarder** 사용자 정의 리소스(CR)는 표준 **Kubernetes** 감사 정책과 동일한 형식을 사용하여 다음과 같은 추가 기능을 제공합니다.

와일드카드

사용자, 그룹, 네임스페이스 및 리소스의 이름에는 선행 또는 후행 * 별표 문자가 있을 수 있습니다. 예를 들어 네임스페이스 `openshift-*` 는 `openshift-apiserver` 또는 `openshift-authentication` 과 일치합니다. 리소스 `*/status` 는 `Pod/status` 또는 `Deployment/status` 와 일치합니다.

기본 규칙

정책의 규칙과 일치하지 않는 이벤트는 다음과 같이 필터링됩니다.

- `get,list,watch` 와 같은 읽기 전용 시스템 이벤트가 삭제됩니다.
- 서비스 계정은 서비스 계정과 동일한 네임스페이스 내에서 발생하는 이벤트를 기록합니다.
- 기타 모든 이벤트는 구성된 속도 제한에 따라 전달됩니다.

이러한 기본값을 비활성화하려면 수준 필드만 있는 규칙으로 규칙 목록을 종료하거나 빈 규칙을 추가합니다.

응답 코드 생략

생략할 정수 상태 코드 목록입니다. 이벤트가 생성되지 않는 HTTP 상태 코드 목록인 **OmitResponseCodes** 필드를 사용하여 응답에서 HTTP 상태 코드를 기반으로 이벤트를 삭제할 수 있습니다. 기본값은 [404, 409, 422, 429]입니다. 값이 빈 목록 []인 경우 상태 코드는 생략되지 않습니다.

ClusterLogForwarder CR 감사 정책은 **OpenShift Container Platform** 감사 정책 외에도 작동합니다. **ClusterLogForwarder CR** 감사 필터는 로그 수집기가 전달하는 내용을 변경하고 동사, 사용자, 그룹, 네임스페이스 또는 리소스별로 필터링하는 기능을 제공합니다. 여러 필터를 생성하여 동일한 감사 스트림의 다른 요약을 다른 위치로 보낼 수 있습니다. 예를 들어 자세한 스트림을 로컬 클러스터 로그 저장소로 전송하고 덜 자세한 스트림을 원격 사이트로 보낼 수 있습니다.



참고

제공된 예제에서는 감사 정책에서 가능한 규칙 범위를 설명하기 위한 것이며 권장되는 구성이 아닙니다.

감사 정책의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: instance
  namespace: openshift-logging
spec:
  pipelines:
    - name: my-pipeline
      inputRefs: audit ①
      filterRefs: my-policy ②
      outputRefs: default
  filters:
    - name: my-policy
      type: kubeAPIAudit
      kubeAPIAudit:
        # Don't generate audit events for all requests in RequestReceived stage.
        omitStages:
          - "RequestReceived"
  rules:
```

```

# Log pod changes at RequestResponse level
- level: RequestResponse
  resources:
    - group: ""
      resources: ["pods"]

# Log "pods/log", "pods/status" at Metadata level
- level: Metadata
  resources:
    - group: ""
      resources: ["pods/log", "pods/status"]

# Don't log requests to a configmap called "controller-leader"
- level: None
  resources:
    - group: ""
      resources: ["configmaps"]
      resourceNames: ["controller-leader"]

# Don't log watch requests by the "system:kube-proxy" on endpoints or services
- level: None
  users: ["system:kube-proxy"]
  verbs: ["watch"]
  resources:
    - group: "" # core API group
      resources: ["endpoints", "services"]

# Don't log authenticated requests to certain non-resource URL paths.
- level: None
  userGroups: ["system:authenticated"]
  nonResourceURLs:
    - "/api*" # Wildcard matching.
    - "/version"

# Log the request body of configmap changes in kube-system.
- level: Request
  resources:
    - group: "" # core API group
      resources: ["configmaps"]
    # This rule only applies to resources in the "kube-system" namespace.
    # The empty string "" can be used to select non-namespaced resources.
    namespaces: ["kube-system"]

# Log configmap and secret changes in all other namespaces at the Metadata level.
- level: Metadata
  resources:
    - group: "" # core API group
      resources: ["secrets", "configmaps"]

# Log all other resources in core and extensions at the Request level.
- level: Request
  resources:
    - group: "" # core API group
    - group: "extensions" # Version of group should NOT be included.

```

```
# A catch-all rule to log all other requests at the Metadata level.
- level: Metadata
```

1

수집되는 로그 유형입니다. 이 필드의 값은 감사 로그, 애플리케이션 로그의 애플리케이션, 인프라 로그용 인프라 또는 애플리케이션에 대해 정의된 이름이 지정된 입력에 대한 감사일 수 있습니다.

2

감사 정책의 이름입니다.

추가 리소스

- 송신 방화벽 및 네트워크 정책 규칙에 대한 로깅

9.4.10. 외부 **Loki** 로깅 시스템으로 로그 전달

기본 로그 저장소에 추가하거나 대신 외부 **Loki** 로깅 시스템으로 로그를 전달할 수 있습니다.

Loki에 대한 로그 전달을 구성하려면 **Loki**에 대한 출력과 출력을 사용하는 파이프라인이 있는 **ClusterLogForwarder** 사용자 정의 리소스(CR)를 생성해야 합니다. **Loki**의 출력은 **HTTP(비보안)** 또는 **HTTPS(보안 HTTP)** 연결을 사용할 수 있습니다.

사전 요구 사항

- CR의 **url** 필드로 지정하는 URL에서 **Loki** 로깅 시스템을 실행해야 합니다.

절차

1.

ClusterLogForwarder CR 오브젝트를 정의하는 **YAML** 파일을 생성하거나 편집합니다.

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
```

```

name: <log_forwarder_name> ①
namespace: <log_forwarder_namespace> ②
spec:
  serviceAccountName: <service_account_name> ③
  outputs:
    - name: loki-insecure ④
      type: "loki" ⑤
      url: http://loki.insecure.com:3100 ⑥
      loki:
        tenantKey: kubernetes.namespace_name
        labelKeys:
          - kubernetes.labels.foo
    - name: loki-secure ⑦
      type: "loki"
      url: https://loki.secure.com:3100
      secret:
        name: loki-secret ⑧
      loki:
        tenantKey: kubernetes.namespace_name ⑨
        labelKeys:
          - kubernetes.labels.foo ⑩
  pipelines:
    - name: application-logs ⑪
      inputRefs: ⑫
        - application
        - audit
      outputRefs: ⑬
        - loki-secure

```

1

레거시 구현에서 CR 이름은 인스턴스 여야 합니다. 다중 로그 전달자 구현에서는 모든 이름을 사용할 수 있습니다.

2

레거시 구현에서 CR 네임스페이스는 **openshift-logging** 이어야 합니다. 다중 로그 전달자 구현에서는 모든 네임스페이스를 사용할 수 있습니다.

3

서비스 계정의 이름입니다. 서비스 계정은 로그 전달자가 **openshift-logging** 네임스페이스에 배포되지 않은 경우에만 다중 로그 전달자 구현에 필요합니다.

4

출력 이름을 지정합니다.

5

유형을 "loki"로 지정합니다.

6

Loki 시스템의 URL 및 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. Loki의 HTTP(S) 통신용 기본 포트는 3100입니다.

7

보안 연결의 경우 secret을 지정하여 인증하는 https 또는 http URL을 지정할 수 있습니다.

8

https 접두사의 경우 TLS 통신의 엔드포인트에 필요한 보안 이름을 지정합니다. 시크릿에는 나타내는 인증서를 가리키는 ca-bundle.crt 키가 포함되어야 합니다. 그렇지 않으면 http 및 https 접두사의 경우 사용자 이름과 암호가 포함된 시크릿을 지정할 수 있습니다. 레거시 구현에서는 openshift-logging 프로젝트에 시크릿이 있어야 합니다. 자세한 내용은 다음 "사용자 이름 및 암호가 포함된 시크릿 설정 예"를 참조하십시오.

9

선택 사항: metadata 키 필드를 지정하여 Loki의 TenantID 필드 값을 생성합니다. 예를 들어 tenantKey: kubernetes.namespace_name을 설정하면 Kubernetes 네임스페이스의 이름이 Loki의 테넌트 ID 값으로 사용됩니다. 지정할 수 있는 다른 로그 레코드 필드를 보려면 다음 "추가 리소스" 섹션의 "로그 레코드 필드" 링크를 참조하십시오.

10

선택 사항: 메타데이터 키 목록을 지정하여 기본 Loki 레이블을 교체합니다. Loki 레이블 이름은 정규식 [a-zA-Z_:][a-zA-Z0-9_]*와 일치해야 합니다. 메타데이터 키의 잘못된 문자는 레이블 이름을 형성하기 위해 _로 교체됩니다. 예를 들어 kubernetes.labels.foo 메타데이터 키는 Loki 레이블 kubernetes_labels_foo 가 됩니다. labelKeys를 설정하지 않으면 기본값은 [log_type, kubernetes.namespace_name, kubernetes.pod_name, kubernetes_host]입니다. Loki는 허용되는 레이블의 크기와 수를 제한하므로 레이블 세트를 작게 유지합니다. [Configuring Loki, limits_config](#)를 참조하십시오. 쿼리 필터를 사용하여 로그 레코드 필드를 기반으로 쿼리할 수 있습니다.

11

선택사항: 파이프라인의 이름을 지정합니다.

12

파이프라인을 사용하여 전달할 로그 유형 (**application**, **infrastructure**, 또는 **audit**)을 지정합니다.

13

이 파이프라인으로 로그를 전달할 때 사용할 출력 이름을 지정합니다.



참고

Loki는 타임스탬프에 의해 로그 스트림을 올바르게 정렬해야 하므로 **labelKeys**에는 항상 **kubernetes_host** 레이블 세트가 포함됩니다. 이렇게 하면 각 스트림이 단일 호스트에서 시작되도록 하여 서로 다른 호스트의 클록 차이로 인해 타임스탬프가 무질서해지는 것을 방지할 수 있습니다.

2.

다음 명령을 실행하여 **ClusterLogForwarder CR** 오브젝트를 적용합니다.

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

추가 리소스

•

[Loki 서버 구성](#)

9.4.11. 외부 Elasticsearch 인스턴스로 로그 전달

내부 로그 저장소에 추가하거나 대신 외부 **Elasticsearch** 인스턴스로 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 생성해야 합니다. 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.



참고

내부 Elasticsearch 인스턴스에만 로그를 전달하려면 ClusterLogForwarder CR을 생성할 필요가 없습니다.

사전 요구 사항

-

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

절차

1.

ClusterLogForwarder CR을 정의하는 YAML 파일을 생성하거나 편집합니다.

ClusterLogForwarder CR의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: <log_forwarder_name> ①
  namespace: <log_forwarder_namespace> ②
spec:
  serviceAccountName: <service_account_name> ③
  outputs:
    - name: elasticsearch-example ④
      type: elasticsearch ⑤
      elasticsearch:
        version: 8 ⑥
        url: http://elasticsearch.example.com:9200 ⑦
        secret:
          name: es-secret ⑧
  pipelines:
    - name: application-logs ⑨
      inputRefs: ⑩
        - application
        - audit
      outputRefs:
        - elasticsearch-example ⑪
        - default ⑫
      labels:
        myLabel: "myValue" ⑬
# ...
```

1

레거시 구현에서 CR 이름은 인스턴스 여야 합니다. 다중 로그 전달자 구현에서는 모든 이름을 사용할 수 있습니다.

2

레거시 구현에서 CR 네임스페이스는 **openshift-logging** 이어야 합니다. 다중 로그 전달자 구현에서는 모든 네임스페이스를 사용할 수 있습니다.

3

서비스 계정의 이름입니다. 서비스 계정은 로그 전달자가 **openshift-logging** 네임스페이스에 배포되지 않은 경우에만 다중 로그 전달자 구현에 필요합니다.

4

출력 이름을 지정합니다.

5

elasticsearch 유형을 지정합니다.

6

Elasticsearch 버전을 지정합니다. **6,7** 또는 **8** 일 수 있습니다.

7

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.

8

https 접두사의 경우 **TLS** 통신의 엔드포인트에 필요한 보안 이름을 지정합니다. 시크릿에는 나타내는 인증서를 가리키는 **ca-bundle.crt** 키가 포함되어야 합니다. 그러지 않으면 **http** 및 **https** 접두사의 경우 사용자 이름과 암호가 포함된 시크릿을 지정할 수 있습니다. 레거시 구현에서는 **openshift-logging** 프로젝트에 시크릿이 있어야 합니다. 자세한 내용은 다음 "사용자 이름 및 암호가 포함된 시크릿 설정 예"를 참조하십시오.

9

선택사항: 파이프라인의 이름을 지정합니다.

10

파이프라인을 사용하여 전달할 로그 유형 (**application**, **infrastructure**, 또는 **audit**)을 지정합니다.

11

이 파이프라인으로 로그를 전달할 때 사용할 출력 이름을 지정합니다.

12

선택사항: **default** 출력을 지정하여 내부 **Elasticsearch** 인스턴스로 로그를 보냅니다.

13

선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

2.

ClusterLogForwarder CR을 적용합니다.

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

예: 사용자 이름과 암호가 포함된 시크릿 설정

사용자 이름과 암호가 포함된 시크릿을 사용하여 외부 **Elasticsearch** 인스턴스에 대한 보안 연결을 인증할 수 있습니다.

예를 들어 타사가 **Elasticsearch** 인스턴스를 작동하기 때문에 상호 TLS(mTLS) 키를 사용할 수 없는 경우 **HTTP** 또는 **HTTPS**를 사용하고 사용자 이름과 암호가 포함된 시크릿을 설정할 수 있습니다.

1.

다음 예와 유사한 **Secret YAML** 파일을 생성합니다. **username** 및 **password** 필드에 **base64**로 인코딩된 값을 사용합니다. 기본적으로 **secret** 유형은 **opaque**입니다.

```
apiVersion: v1
kind: Secret
metadata:
  name: openshift-test-secret
```

```

data:
  username: <username>
  password: <password>
# ...

```

2. 시크릿을 생성합니다.

```
$ oc create secret -n openshift-logging openshift-test-secret.yaml
```

3. ClusterLogForwarder CR에서 시크릿 이름을 지정합니다.

```

kind: ClusterLogForwarder
metadata:
  name: instance
  namespace: openshift-logging
spec:
  outputs:
    - name: elasticsearch
      type: "elasticsearch"
      url: https://elasticsearch.secure.com:9200
      secret:
        name: openshift-test-secret
# ...

```



참고

url 필드의 값에서 접두사는 http 또는 https가 될 수 있습니다.

4. CR 오브젝트를 적용합니다.

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

9.4.12. Fluentd 정방향 프로토콜을 사용하여 로그 전달

Fluentd 전달 프로토콜을 사용하여 기본 Elasticsearch 로그 저장소 대신 또는 기본 Elasticsearch 로그 저장소 외에 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 생성해야 합니다.

Fluentd 출력은 TCP(비보안) 또는 TLS(보안 TCP) 연결을 사용할 수 있습니다.

사전 요구 사항

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

절차

- ClusterLogForwarder CR 오브젝트를 정의하는 YAML 파일을 생성하거나 편집합니다.

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

①

ClusterLogForwarder CR의 이름은 `instance`여야 합니다.

2

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

3

출력 이름을 지정합니다.

4

fluentdForward 유형을 지정합니다.

5

유효한 절대 URL로 외부 Fluentd 인스턴스의 URL 및 포트를 지정합니다. **tcp**(비보안) 또는 **tls**(보안 TCP) 프로토콜을 사용할 수 있습니다. CIDR 주석을 사용하는 클러스터 전체 프록시가 활성화된 경우 출력은 IP 주소가 아닌 서버 이름 또는 FQDN이어야 합니다.

6

tls 접두사를 사용하는 경우 TLS 통신을 위해 끝점에 필요한 시크릿 이름을 지정해야 합니다. 시크릿은 **openshift-logging** 프로젝트에 있어야 하며 나타내는 인증서를 가리키는 **ca-bundle.crt** 키가 포함되어야 합니다.

7

선택사항: 파일라인의 이름을 지정합니다.

8

파일라인을 사용하여 전달할 로그 유형 (**application**, **infrastructure**, 또는 **audit**)을 지정합니다.

9

이 파일라인으로 로그를 전달할 때 사용할 출력 이름을 지정합니다.

10

선택사항: **default** 출력을 지정하여 내부 **Elasticsearch** 인스턴스로 로그를 전달합니다.

11

선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

12

선택사항: 지원되는 유형의 다른 외부 로그 집계에 로그를 전달하도록 여러 출력을 구성합니다.

- 선택 사항: 문자열을 설명하는 이름입니다.
- **inputRefs**는 **pipeline: application, infrastructure** 또는 **audit**를 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
```

9.4.12.1. Logstash가 fluentd에서 데이터를 수집할 수 있도록 nanosecond precision 사용

fluentd에서 로그 데이터를 수집하려면 **Logstash** 구성 파일에서 **nanosecond precision**을 활성화해야 합니다.

절차

-

Logstash 설정 파일에서 **nanosecond_precision**을 **true**로 설정합니다.

Logstash 구성 파일 예

```
input { tcp { codec => fluent { nanosecond_precision => true } port => 24114 } }
filter { }
```

```
output { stdout { codec => rubydebug } }
```

9.4.13. 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 수집기에 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 생성해야 합니다. syslog 출력은 UDP, TCP 또는 TLS 연결을 사용할 수 있습니다.

사전 요구 사항

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

절차

1.

ClusterLogForwarder CR 오브젝트를 정의하는 YAML 파일을 생성하거나 편집합니다.

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: <log_forwarder_name> ①
  namespace: <log_forwarder_namespace> ②
spec:
  serviceAccountName: <service_account_name> ③
  outputs:
    - name: rsyslog-east ④
      type: syslog ⑤
      syslog:
        facility: local0
        rfc: RFC3164
        payloadKey: message
        severity: informational
      url: 'tls://rsyslogserver.east.example.com:514' ⑦
      secret:
        name: syslog-secret ⑧
```

```

- name: rsyslog-west
  type: syslog
  syslog:
    appName: myapp
    facility: user
    msgID: mymsg
    procID: myproc
    rfc: RFC5424
    severity: debug
  url: 'tcp://rsyslogserver.west.example.com:514'
pipelines:
- name: syslog-east 9
  inputRefs: 10
  - audit
  - application
  outputRefs: 11
  - rsyslog-east
  - default 12
  labels:
    secure: "true" 13
    syslog: "east"
- name: syslog-west 14
  inputRefs:
  - infrastructure
  outputRefs:
  - rsyslog-west
  - default
  labels:
    syslog: "west"

```

1

레거시 구현에서 CR 이름은 인스턴스 여야 합니다. 다중 로그 전달자 구현에서는 모든 이름을 사용할 수 있습니다.

2

레거시 구현에서 CR 네임스페이스는 **openshift-logging** 이어야 합니다. 다중 로그 전달자 구현에서는 모든 네임스페이스를 사용할 수 있습니다.

3

서비스 계정의 이름입니다. 서비스 계정은 로그 전달자가 **openshift-logging** 네임스페이스에 배포되지 않은 경우에만 다중 로그 전달자 구현에 필요합니다.

4

출력 이름을 지정합니다.

5

6

선택 사항: 아래에 나열된 **syslog** 매개변수를 지정합니다.

7

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 **FQDN**이어야 합니다.

8

tls 접두사를 사용하는 경우 **TLS** 통신을 위해 끝점에서 요구하는 시크릿 이름을 지정해야 합니다. 시크릿에는 나타내는 인증서를 가리키는 **ca-bundle.crt** 키가 포함되어야 합니다. 레거시 구현에서는 **openshift-logging** 프로젝트에 시크릿이 있어야 합니다.

9

선택사항: 파이프라인의 이름을 지정합니다.

10

파이프라인을 사용하여 전달할 로그 유형 (**application**, **infrastructure**, 또는 **audit**)을 지정합니다.

11

이 파이프라인으로 로그를 전달할 때 사용할 출력 이름을 지정합니다.

12

선택사항: **default** 출력을 지정하여 내부 **Elasticsearch** 인스턴스로 로그를 전달합니다.

13

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

14

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

2.

CR 오브젝트를 생성합니다.

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

9.4.13.1. 메시지 출력에 로그 소스 정보 추가

ClusterLogForwarder 사용자 정의 리소스(CR)에 **AddLogSource** 필드를 추가하여 **namespace_name,pod_name** 요소를 레코드의 **message** 필드에 추가할 수 있습니다.

```
spec:
  outputs:
    - name: syslogout
      syslog:
        addLogSource: true
        facility: user
        payloadKey: message
        rfc: RFC3164
        severity: debug
        tag: mytag
        type: syslog
        url: tls://syslog-receiver.openshift-logging.svc:24224
  pipelines:
    - inputRefs:
        - application
      name: test-app
      outputRefs:
        - syslogout
```



참고

이 구성은 [RFC3164](#) 및 [RFC5424](#)와 모두 호환됩니다.

AddLogSource가 없는 syslog 메시지 출력 예

```
<15>1 2020-11-15T17:06:14+00:00 fluentd-9hkb4 mytag --- {"msgcontent"=>"Message Contents", "timestamp"=>"2020-11-15 17:06:09", "tag_key"=>"rec_tag", "index"=>56}
```

AddLogSource의 syslog 메시지 출력 예

```
<15>1 2020-11-16T10:49:37+00:00 crc-j55b9-master-0 mytag --- namespace_name=clo-test-6327,pod_name=log-generator-ff9746c49-qxm7l,container_name=log-generator,message={"msgcontent":"My life is my message", "timestamp":"2020-11-16 10:49:36", "tag_key":"rec_tag", "index":76}
```

9.4.13.2. Syslog 매개변수

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

- 시설: [syslog facility](#). 값은 10진수 정수 또는 대소문자를 구분하지 않는 키워드일 수 있습니다.
 - 커널 메시지의 경우 0 또는 kern
 - 사용자 수준 메시지의 경우 1 또는 user, 기본값입니다.

- 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**
- 선택 사항: **payloadKey: syslog** 메시지의 페이로드로 사용할 레코드 펠드입니다.



참고

payloadKey 매개변수를 구성하면 다른 매개 변수가 **syslog**로 전달되지 않습니다.

- **rfc: syslog**를 사용하여 로그를 보내는 데 사용할 RFC입니다. 기본값은 **RFC5424**입니다.
- **심각도:** 발신되는 **syslog** 레코드에 설정할 **syslog** 심각도입니다. 값은 10진수 정수 또는 대소문자를 구분하지 않는 키워드일 수 있습니다.
- 시스템을 사용할 수 없음을 나타내는 메시지의 경우 **0** 또는 **Emergency**
- 조치를 즉시 취해야 함을 나타내는 메시지의 경우 **1** 또는 **Alert**
- 위험 상태를 나타내는 메시지의 경우 **2** 또는 **Critical**
- 오류 상태를 나타내는 메시지의 경우 **3** 또는 **Error**
- 경고 조건을 나타내는 메시지의 경우 **4** 또는 **Warning**
- 정상이지만 중요한 조건을 나타내는 메시지의 경우 **5** 또는 **Notice**

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

9.4.13.3. 추가 RFC5424 syslog 매개변수

RFC5424에는 다음 매개변수가 적용됩니다.

- **appName**: APP-NAME은 로그를 보낸 애플리케이션을 식별하는 자유 텍스트 문자열입니다. RFC5424에 대해 지정해야 합니다.
- **msgID**: MSGID는 메시지 유형을 식별하는 자유 텍스트 문자열입니다. RFC5424에 대해 지정해야 합니다.
- **procID**: PROCID는 자유 텍스트 문자열입니다. 값이 변경되면 **syslog** 보고가 중단되었음을 나타냅니다. RFC5424에 대해 지정해야 합니다.

9.4.14. Kafka 브로커로 로그 전달

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

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

절차

1.

ClusterLogForwarder CR 오브젝트를 정의하는 YAML 파일을 생성하거나 편집합니다.

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: <log_forwarder_name> ①
  namespace: <log_forwarder_namespace> ②
spec:
  serviceAccountName: <service_account_name> ③
  outputs:
    - name: app-logs ④
      type: kafka ⑤
      url: tls://kafka.example.devlab.com:9093/app-topic ⑥
      secret:
        name: kafka-secret ⑦
    - name: infra-logs
      type: kafka
      url: tcp://kafka.devlab2.example.com:9093/infra-topic ⑧
    - name: audit-logs
      type: kafka
      url: tls://kafka.qelab.example.com:9093/audit-topic
      secret:
        name: kafka-secret-qe
  pipelines:
    - name: app-topic ⑨
      inputRefs: ⑩
      - application
      outputRefs: ⑪
      - app-logs
      labels:
        logType: "application" ⑫
    - name: infra-topic ⑬
      inputRefs:
        - infrastructure
      outputRefs:
        - infra-logs
      labels:
        logType: "infra"
    - name: audit-topic
      inputRefs:
        - audit
      outputRefs:
        - audit-logs
      labels:
        logType: "audit"
```

①

레거시 구현에서 CR 이름은 인스턴스 여야 합니다. 다중 로그 전달자 구현에서는 모든 이름을 사용할 수 있습니다.

2

레거시 구현에서 CR 네임스페이스는 **openshift-logging** 이어야 합니다. 다중 로그 전달자 구현에서는 모든 네임스페이스를 사용할 수 있습니다.

3

서비스 계정의 이름입니다. 서비스 계정은 로그 전달자가 **openshift-logging** 네임스페이스에 배포되지 않은 경우에만 다중 로그 전달자 구현에 필요합니다.

4

출력 이름을 지정합니다.

5

kafka 유형을 지정합니다.

6

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.

7

tls 접두사를 사용하는 경우 TLS 통신을 위해 끝점에 필요한 시크릿 이름을 지정해야 합니다. 시크릿에는 나타내는 인증서를 가리키는 **ca-bundle.crt** 키가 포함되어야 합니다. 레거시 구현에서는 **openshift-logging** 프로젝트에 시크릿이 있어야 합니다.

8

선택 사항: 비보안 출력을 보내려면 URL 앞에 있는 **tcp** 접두사를 사용합니다. 이 출력에서 **secret** 키와 해당 **name**을 생략합니다.

9

선택사항: 파일라인의 이름을 지정합니다.

10

파일라인을 사용하여 전달할 로그 유형 (**application**, **infrastructure**, 또는 **audit**)을 지정합니다.

11

이 파이프라인으로 로그를 전달할 때 사용할 출력 이름을 지정합니다.

12

선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

13

선택사항: 지원되는 유형의 다른 외부 로그 집계에 로그를 전달하도록 여러 출력을 구성합니다.

- 파이프라인을 설명하는 이름입니다.
- **inputRefs**는 pipeline: application, infrastructure 또는 audit를 사용하여 전달 할 로그 유형입니다.
- **outputRefs**는 사용할 출력의 이름입니다.
- 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

2.

선택 사항: 단일 출력을 여러 Kafka 브로커로 전달하려면 다음 예와 같이 Kafka 브로커 배열 을 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

2

brokers 키를 사용하여 하나 이상의 브로커 배열을 지정합니다.

3

topic 키를 사용하여 로그를 수신하는 대상 주제를 지정합니다.

3.

다음 명령을 실행하여 **ClusterLogForwarder CR**을 적용합니다.

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

9.4.15. Amazon CloudView로 로그 전달

AWS(Amazon Web Services)에서 호스팅하는 모니터링 및 로그 스토리지 서비스인 **Amazon CloudMonitor**에 로그를 전달할 수 있습니다. 기본 로그 저장소에 추가하거나 대신 **logs**를 **NetNamespace**로 전달할 수 있습니다.

CloudMonitor에 대한 로그 전달을 구성하려면 **CloudMonitor**의 출력이 있는 **ClusterLogForwarder** 사용자 정의 리소스(CR)와 출력을 사용하는 파이프라인을 생성해야 합니다.

절차

1.

aws_access_key_id 및 **aws_secret_access_key** 필드를 사용하여 **base64**로 인코딩된 AWS 인증 정보를 지정하는 **Secret YAML** 파일을 만듭니다. 예를 들면 다음과 같습니다.

```
apiVersion: v1
kind: Secret
metadata:
  name: cw-secret
  namespace: openshift-logging
data:
  aws_access_key_id: QUtJQUIPU0ZPRE5ON0VYQU1QTEUK
  aws_secret_access_key:
d0phbHJYVXRuRkVNSS9LN01ERU5HL2JQeFJmaUNZRVhBTVBMRUtFWQo=
```

2.

시크릿을 생성합니다. 예를 들면 다음과 같습니다.

```
$ oc apply -f cw-secret.yaml
```

3.

ClusterLogForwarder CR 오브젝트를 정의하는 YAML 파일을 생성하거나 편집합니다. 파일에서 시크릿 이름을 지정합니다. 예를 들면 다음과 같습니다.

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: <log_forwarder_name> ①
  namespace: <log_forwarder_namespace> ②
spec:
  serviceAccountName: <service_account_name> ③
  outputs:
    - name: cw ④
      type: cloudwatch ⑤
      cloudwatch:
        groupBy: logType ⑥
        groupPrefix: <group prefix> ⑦
        region: us-east-2 ⑧
      secret:
        name: cw-secret ⑨
  pipelines:
    - name: infra-logs ⑩
      inputRefs: ⑪
        - infrastructure
        - audit
        - application
      outputRefs:
        - cw ⑫
```

①

레거시 구현에서 CR 이름은 인스턴스 여야 합니다. 다중 로그 전달자 구현에서는 모든 이름을 사용할 수 있습니다.

②

레거시 구현에서 CR 네임스페이스는 **openshift-logging** 이어야 합니다. 다중 로그 전달자 구현에서는 모든 네임스페이스를 사용할 수 있습니다.

③

서비스 계정의 이름입니다. 서비스 계정은 로그 전달자가 **openshift-logging** 네임스페이스에 배포되지 않은 경우에만 다중 로그 전달자 구현에 필요합니다.

④

출력 이름을 지정합니다.

5

cloudwatch 유형을 지정합니다.

6

선택 사항: 로그를 그룹화하는 방법을 지정합니다.

- **logType** 은 각 로그 유형에 대한 로그 그룹을 생성합니다.
- **namespaceName**은 각 애플리케이션 네임 스페이스에 대한 로그 그룹을 생성합니다. 인프라 및 감사 로그를 위해 별도의 로그 그룹도 생성합니다.
- **namespaceUUID**는 각 애플리케이션 네임스페이스 UUID에 대한 새 로그 그룹을 생성합니다. 인프라 및 감사 로그를 위해 별도의 로그 그룹도 생성합니다.

7

선택 사항: 로그 그룹 이름으로 기본 **infrastructureName** 접두사를 바꾸려면 문자열을 지정합니다.

8

AWS 리전을 지정합니다.

9

AWS 인증 정보가 포함된 시크릿의 이름을 지정합니다.

10

선택사항: 파이프라인의 이름을 지정합니다.

11

파이프라인을 사용하여 전달할 로그 유형 (**application**, **infrastructure**, 또는 **audit**)을 지정합니다.

12

이 파일프라인으로 로그를 전달할 때 사용할 출력 이름을 지정합니다.

4.

CR 오브젝트를 생성합니다.

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

예 : Amazon Cloud Watch에서 ClusterLogForwarder 사용

여기서 ClusterLogForwarder 사용자 정의 리소스(CR)의 예와 Amazon CloudMonitor로 출력되는 로그 데이터가 표시됩니다.

mycluster라는 OpenShift Container Platform 클러스터를 실행하고 있다고 가정합니다. 다음 명령은 나중에 aws 명령을 구성하는 데 사용할 클러스터의 infrastructureName을 반환합니다.

```
$ oc get Infrastructure/cluster -ojson | jq .status.infrastructureName
"mycluster-7977k"
```

이 예제에 대한 로그 데이터를 생성하려면 app이라는 네임스페이스에서 busybox Pod를 실행합니다. busybox Pod는 3초마다 stdout에 메시지를 씁니다.

```
$ oc run busybox --image=busybox -- sh -c 'while true; do echo "My life is my message"; sleep 3; done'
$ oc logs -f busybox
My life is my message
My life is my message
My life is my message
...
...
```

busybox Pod가 실행 되는 앱 네임스페이스의 UUID를 조회할 수 있습니다.

```
$ oc get ns/app -ojson | jq .metadata.uid
"794e1e1a-b9f5-4958-a190-e76a9b53d7bf"
```

ClusterLogForwarder 사용자 정의 리소스(CR)에서 infrastructure, audit, application 로그 유형을 all-logs 파일프라인에 대한 입력으로 구성합니다. 또한 이 파일프라인을 cw 출력에 연결하여 us-east-2 리전의 CloudMonitor 인스턴스로 로그를 전달합니다.

```

apiVersion: "logging.openshift.io/v1"
kind: ClusterLogForwarder
metadata:
  name: instance
  namespace: openshift-logging
spec:
  outputs:
    - name: cw
      type: cloudwatch
      cloudwatch:
        groupBy: logType
        region: us-east-2
        secret:
          name: cw-secret
  pipelines:
    - name: all-logs
      inputRefs:
        - infrastructure
        - audit
        - application
      outputRefs:
        - cw

```

CloudMonitor의 각 리전에는 세 가지 수준의 오브젝트가 포함되어 있습니다.

- 로그 그룹
- 로그 스트림
- 로그 이벤트

ClusterLogForwarding CR의 **groupBy: logType** 을 사용하는 경우 **inputRefs** 의 세 가지 로그 유형은 Amazon Cloudwatch에 3개의 로그 그룹을 생성합니다.

```
$ aws --output json logs describe-log-groups | jq .logGroups[].logGroupName
"mycluster-7977k.application"
"mycluster-7977k.audit"
"mycluster-7977k.infrastructure"
```

각 로그 그룹에는 로그 스트림이 포함되어 있습니다.

```
$ aws --output json logs describe-log-streams --log-group-name mycluster-7977k.application |
```

```
jq .logStreams[].logStreamName
"kubernetes.var.log.containers.busybox_app_busybox-
da085893053e20beddd6747acdbaf98e77c37718f85a7f6a4facf09ca195ad76.log"
```

```
$ aws --output json logs describe-log-streams --log-group-name mycluster-7977k.audit | jq
.logStreams[].logStreamName
"ip-10-0-131-228.us-east-2.compute.internal.k8s-audit.log"
"ip-10-0-131-228.us-east-2.compute.internal.linux-audit.log"
"ip-10-0-131-228.us-east-2.compute.internal.openshift-audit.log"
..."
```

```
$ aws --output json logs describe-log-streams --log-group-name mycluster-
7977k.infrastructure | jq .logStreams[].logStreamName
"ip-10-0-131-228.us-east-2.compute.internal.kubernetes.var.log.containers.apiserver-
69f9fd9b58-zqzw5_openshift-oauth-apiserver_oauth-apiserver-
453c5c4ee026fe20a6139ba6b1cdd1bed25989c905bf5ac5ca211b7cbb5c3d7b.log"
"ip-10-0-131-228.us-east-2.compute.internal.kubernetes.var.log.containers.apiserver-
797774f7c5-lftrx_openshift-apiserver_openshift-apiserver-
ce51532df7d4e4d5f21c4f4be05f6575b93196336be0027067fd7d93d70f66a4.log"
"ip-10-0-131-228.us-east-2.compute.internal.kubernetes.var.log.containers.apiserver-
797774f7c5-lftrx_openshift-apiserver_openshift-apiserver-check-endpoints-
82a9096b5931b5c3b1d6dc4b66113252da4a6472c9fff48623baee761911a9ef.log"
..."
```

각 로그 스트림에는 로그 이벤트가 포함되어 있습니다. **busybox Pod**에서 로그 이벤트를 보려면 **application** 로그 그룹에서 로그 스트림을 지정합니다.

```
$ aws logs get-log-events --log-group-name mycluster-7977k.application --log-stream-name
kubernetes.var.log.containers.busybox_app_busybox-
da085893053e20beddd6747acdbaf98e77c37718f85a7f6a4facf09ca195ad76.log
{
  "events": [
    {
      "timestamp": 1629422704178,
      "message": "{\"docker\":
{\"container_id\":\"da085893053e20beddd6747acdbaf98e77c37718f85a7f6a4facf09ca195ad76\"},
\"kubernetes\":
{\"container_name\":\"busybox\", \"namespace_name\":\"app\", \"pod_name\":\"busybox\", \"co
ntainer_image\":\"docker.io/library/busybox:latest\", \"container_image_id\":\"docker.io/library/
busybox@sha256:0f354ec1728d9ff32edcd7d1b8bbdfc798277ad36120dc3dc683be44524c8b60\",
\"pod_id\":\"870be234-90a3-4258-b73f-4f4d6e2777c7\", \"host\":\"ip-10-0-216-3.us-east-
2.compute.internal\", \"labels\":
{\"run\":\"busybox\"}, \"master_url\":\"https://kubernetes.default.svc\", \"namespace_id\":\"794e
1e1a-b9f5-4958-a190-e76a9b53d7bf\", \"namespace_labels\":
{\"kubernetes_io/metadata_name\":\"app\"}}, \"message\":\"My life is my
message\", \"level\":\"unknown\", \"hostname\":\"ip-10-0-216-3.us-east-
2.compute.internal\", \"pipeline_metadata\":{\"collector\":
{\"ipaddr4\":\"10.0.216.3\", \"inputname\":\"fluent-plugin-
systemd\", \"name\":\"fluentd\", \"received_at\":\"2021-08-
20T01:25:08.085760+00:00\", \"version\":\"1.7.4 1.6.0\"}, \"@timestamp\":\"2021-08-
20T01:25:04.178986+00:00\", \"viaq_index_name\":\"app-
write\", \"viaq_msg_id\":\"NWRjZmUyMWQtZjgzNC00MjI4LTk3MjMtNTk3NmY3ZjU4NDk1\", \"log
_type\":\"application\", \"time\":\"2021-08-20T01:25:04+00:00\"},
```

```

    "ingestionTime": 1629422744016
},
...

```

예: 로그 그룹 이름에 접두사 사용자 지정

로그 그룹 이름에서는 기본 **infrastructureName** 접두사인 **mycluster-7977k**를 **demo-group-prefix**와 같은 임의의 문자열로 바꿀 수 있습니다. 이 변경을 수행하려면 **ClusterLogForwarding CR**에서 **groupPrefix** 필드를 업데이트합니다.

```

cloudwatch:
  groupBy: logType
  groupPrefix: demo-group-prefix
  region: us-east-2

```

groupPrefix 값은 기본 **infrastructureName** 접두사를 대체합니다.

```
$ aws --output json logs describe-log-groups | jq .logGroups[].logGroupName
"demo-group-prefix.application"
"demo-group-prefix.audit"
"demo-group-prefix.infrastructure"
```

예: 애플리케이션 네임스페이스 이름 뒤에 로그 그룹 이름 지정

클러스터의 각 애플리케이션 네임스페이스에 대해 애플리케이션 네임스페이스 이름을 기반으로 하는 로그 그룹을 **CloudWatch**에서 생성할 수 있습니다.

애플리케이션 네임스페이스 오브젝트를 삭제하고 이름이 같은 새 항목을 생성하면 **CloudMonitor**는 이전과 동일한 로그 그룹을 계속 사용합니다.

서로 동일한 이름이 있는 연속적인 애플리케이션 네임스페이스 오브젝트를 고려하는 경우 이 예제에 설명된 접근 방식을 사용합니다. 또는 결과 로그 그룹을 서로 구분해야 하는 경우 대신 다음 "애플리케이션 네임스페이스 UUID에 대한 로그 그룹 설정" 섹션을 참조하십시오.

애플리케이션 네임스페이스의 이름을 기반으로 이름이 인 애플리케이션 로그 그룹을 생성하려면 **ClusterLogForwarder CR**에서 **groupBy** 필드의 값을 **namespaceName**으로 설정합니다.

```

cloudwatch:
  groupBy: namespaceName
  region: us-east-2

```

groupBy를 **namespaceName**으로 설정하면 애플리케이션 로그 그룹에만 영향을 미칩니다. **audit** 및 **infrastructure** 로그 그룹에는 영향을 미치지 않습니다.

Amazon Cloudwatch에서 각 로그 그룹 이름 끝에 네임스페이스 이름이 표시됩니다. 단일 애플리케이션 네임스페이스인 "app"이 있으므로 다음 출력에서는 **mycluster-7977k.application** 대신 새 **mycluster-7977k.app** 로그 그룹이 표시됩니다.

```
$ aws --output json logs describe-log-groups | jq .logGroups[].logGroupName
"mycluster-7977k.app"
"mycluster-7977k.audit"
"mycluster-7977k.infrastructure"
```

이 예제의 클러스터에 여러 애플리케이션 네임스페이스가 포함된 경우 출력에 각 네임스페이스에 하나씩 여러 개의 로그 그룹이 표시됩니다.

groupBy 필드는 애플리케이션 로그 그룹에만 영향을 미칩니다. **audit** 및 **infrastructure** 로그 그룹에는 영향을 미치지 않습니다.

예: 애플리케이션 네임스페이스 UUID 후 로그 그룹 이름 지정

클러스터의 각 애플리케이션 네임스페이스에 대해 애플리케이션 네임스페이스 UUID를 기반으로 하는 로그 그룹을 CloudWatch에서 생성할 수 있습니다.

애플리케이션 네임스페이스 오브젝트를 삭제하고 새 오브젝트를 생성하면 CloudMonitor에서 새 로그 그룹을 생성합니다.

동일한 이름을 가진 연속적인 애플리케이션 네임스페이스 개체를 서로 다른 것으로 간주하는 경우 이 예에서 설명하는 접근 방식을 사용하십시오. 또는 이전의 "애플리케이션 네임스페이스 이름에 대한 로그 그룹 이름 지정" 섹션을 참조하십시오.

애플리케이션 네임스페이스 UUID 후에 로그 그룹의 이름을 지정하려면 ClusterLogForwarder CR에서 **groupBy** 필드의 값을 **namespaceUUID**로 설정합니다.

```
cloudwatch:
  groupBy: namespaceUUID
  region: us-east-2
```

Amazon Cloudwatch에서 각 로그 그룹 이름 끝에 네임스페이스 UUID가 표시됩니다. 단일 애플리케이션 네임스페이스인 "app"이 있으므로 다음 출력에서는 **mycluster-7977k.application** 대신 새로운 **mycluster-7977k.794e1e1a-b9f5-4958-a190-e76a9b53d7bf** 로그 그룹이 표시됩니다.

```
$ aws --output json logs describe-log-groups | jq .logGroups[].logGroupName
"mycluster-7977k.794e1e1a-b9f5-4958-a190-e76a9b53d7bf" // uid of the "app" namespace
"mycluster-7977k.audit"
"mycluster-7977k.infrastructure"
```

groupBy 필드는 애플리케이션 로그 그룹에만 영향을 미칩니다. **audit** 및 **infrastructure** 로그 그룹에는 영향을 미치지 않습니다.

9.4.15.1. 기존 AWS 역할을 사용하여 AWSIdentityProvider의 시크릿 생성

AWS에 대한 기존 역할이 있는 경우 **oc create secret --from-literal** 명령을 사용하여 STS로 AWS의 시크릿을 생성할 수 있습니다.

절차

- CLI에서 다음을 입력하여 AWS의 시크릿을 생성합니다.

```
$ oc create secret generic cw-sts-secret -n openshift-logging --from-literal=role_arn=arn:aws:iam::123456789012:role/my-role_with-permissions
```

보안 예

```
apiVersion: v1
kind: Secret
metadata:
  namespace: openshift-logging
  name: my-secret-name
stringData:
  role_arn: arn:aws:iam::123456789012:role/my-role_with-permissions
```

9.4.16. STS 활성화된 클러스터에서 Amazon STS로 로그 전달

AWS STS(Security Token Service)가 활성화된 클러스터의 경우 **CCO(Cloud Credential Operator)** 유틸리티 **ccocctl** 을 사용하여 AWS 서비스 계정을 수동으로 생성하거나 인증 정보 요청을 생성할 수 있습니다.

사전 요구 사항

- Red Hat OpenShift에 대한 로깅: 5.5 이상

절차

- 아래 템플릿을 사용하여 **CredentialsRequest** 사용자 정의 리소스 **YAML**을 생성합니다.

ECDHE 인증 정보 요청 템플릿

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: <your_role_name>-credrequest
  namespace: openshift-cloud-credential-operator
spec:
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: AWSProviderSpec
    statementEntries:
      - action:
          - logs:PutLogEvents
          - logs>CreateLogGroup
          - logs:PutRetentionPolicy
          - logs>CreateLogStream
          - logs:DescribeLogGroups
          - logs:DescribeLogStreams
        effect: Allow
        resource: arn:aws:logs:*:*/
  secretRef:
    name: <your_role_name>
    namespace: openshift-logging
  serviceAccountNames:
    - logcollector
```

- ccocctl 명령 을 사용하여 **CredentialsRequest CR**을 사용하여 AWS에 대한 역 할을 생성합니다. **CredentialsRequest** 오브젝트를 사용하면 이 ccocctl 명령은 지정된 OIDC ID 공급자에 연

결된 신뢰 정책과 NetNamespace 리소스에 대한 작업을 수행할 수 있는 권한을 부여하는 권한 정책을 사용하여 IAM 역할을 생성합니다. 이 명령은 /<path_to_ccctl_output_dir>/manifests/openshift-logging-<your_role_name>-credentials.yaml에 YAML 구성 파일도 생성합니다. 이 시크릿 파일에는 AWS IAM ID 공급자의 인증 중에 사용되는 role_arn 키/값이 포함되어 있습니다.

```
$ ccctl aws create-iam-roles \
--name=<name> \
--region=<aws_region> \
--credentials-requests-dir=
<path_to_directory_with_list_of_credentials_requests>/credrequests \
--identity-provider-arn=arn:aws:iam::<aws_account_id>:oidc-provider/<name>-
oidc.s3.<aws_region>.amazonaws.com ①
```

1

<name>은 클라우드 리소스에 태그하는 데 사용되는 이름이며 STS 클러스터 설치 중에 사용되는 이름과 일치해야 합니다.

3.

생성된 보안을 적용합니다.

```
$ oc apply -f output/manifests/openshift-logging-<your_role_name>-credentials.yaml
```

4.

ClusterLogForwarder 사용자 정의 리소스를 생성하거나 편집합니다.

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: <log_forwarder_name> ①
  namespace: <log_forwarder_namespace> ②
spec:
  serviceAccountName: clf-collector ③
  outputs:
    - name: cw ④
      type: cloudwatch ⑤
      cloudwatch:
        groupBy: logType ⑥
        groupPrefix: <group prefix> ⑦
        region: us-east-2 ⑧
      secret:
        name: <your_secret_name> ⑨
  pipelines:
    - name: to-cloudwatch ⑩
      inputRefs: ⑪
        - infrastructure
        - audit
```

- application
outputRefs:
- cw 12

1

레거시 구현에서 CR 이름은 인스턴스 여야 합니다. 다중 로그 전달자 구현에서는 모든 이름을 사용할 수 있습니다.

2

레거시 구현에서 CR 네임스페이스는 **openshift-logging** 이어야 합니다. 다중 로그 전달자 구현에서는 모든 네임스페이스를 사용할 수 있습니다.

3

clf-collector 서비스 계정을 지정합니다. 서비스 계정은 로그 전달자가 **openshift-logging** 네임스페이스에 배포되지 않은 경우에만 다중 로그 전달자 구현에 필요합니다.

4

출력 이름을 지정합니다.

5

cloudwatch 유형을 지정합니다.

6

선택 사항: 로그를 그룹화하는 방법을 지정합니다.

- **logType** 은 각 로그 유형에 대한 로그 그룹을 생성합니다.
- **namespaceName**은 각 애플리케이션 네임 스페이스에 대한 로그 그룹을 생성합니다. 인프라 및 감사 로그는 영향을 받지 않으며 **logType** 으로 그룹화됩니다.
- **namespaceUUID**는 각 애플리케이션 네임스페이스 UUID에 대한 새 로그 그룹을 생성합니다. 인프라 및 감사 로그를 위해 별도의 로그 그룹도 생성합니다.

7

선택 사항: 로그 그룹 이름으로 기본 **infrastructureName** 접두사를 바꾸려면 문자열을 지정합니다.

8

AWS 리전을 지정합니다.

9

AWS 인증 정보가 포함된 시크릿의 이름을 지정합니다.

10

선택사항: 파이프라인의 이름을 지정합니다.

11

파이프라인을 사용하여 전달할 로그 유형 (**application**, **infrastructure**, 또는 **audit**)을 지정합니다.

12

이 파이프라인으로 로그를 전달할 때 사용할 출력 이름을 지정합니다.

추가 리소스



[AWS STS API Reference](#)

9.5. 로깅 수집기 구성

Red Hat OpenShift의 로깅은 클러스터에서 작업 및 애플리케이션 로그를 수집하고 **Kubernetes Pod** 및 프로젝트 메타데이터로 데이터를 강화합니다. **ClusterLogging** 사용자 정의 리소스(CR)의 **spec.collection** 스템자를 통해 로그 수집기에 대한 지원되는 모든 수정을 수행할 수 있습니다.

9.5.1. 로그 수집기 구성

ClusterLogging 사용자 정의 리소스(CR)를 수정하여 로깅에서 사용하는 로그 수집기 유형을 구성할 수 있습니다.



참고

Fluentd는 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다. **Red Hat**은 현재 릴리스 라이프사이클 동안 이 기능에 대한 버그 수정 및 지원을 제공하지만 이 기능은 더 이상 개선 사항을 받지 않습니다. **Fluentd** 대신 **Vector**를 사용할 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- **OpenShift CLI(oc)**가 설치되어 있습니다.
- **Red Hat OpenShift Logging Operator**가 설치되어 있습니다.
- **ClusterLogging CR**을 생성했습니다.

절차

1. **ClusterLogging CR** 컬렉션 사양을 수정합니다.

ClusterLogging CR 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
# ...
spec:
# ...
collection:
  type: <log_collector_type> ①
  resources: {}
  tolerations: {}
# ...
```

①

2.

다음 명령을 실행하여 **ClusterLogging CR**을 적용합니다.

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

9.5.2. LogFileMetricExporter 리소스 생성

로깅 버전 5.8 이상 버전에서는 **LogFileMetricExporter**는 기본적으로 수집기와 함께 더 이상 배포되지 않습니다. 컨테이너를 실행하여 생성된 로그에서 메트릭을 생성하려면 **LogFileMetricExporter CR**(사용자 정의 리소스)을 수동으로 생성해야 합니다.

LogFileMetricExporter CR을 생성하지 않으면 OpenShift Container Platform 웹 콘솔 대시보드에 **Produced Logs** 의 **No datapoints found** 메시지가 표시될 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- Red Hat OpenShift Logging Operator가 설치되어 있습니다.
- OpenShift CLI(oc)가 설치되어 있습니다.

절차

1.

LogFileMetricExporter CR을 YAML 파일로 생성합니다.

LogFileMetricExporter CR 예

```
apiVersion: logging.openshift.io/v1alpha1
kind: LogFileMetricExporter
metadata:
  name: instance
  namespace: openshift-logging
spec:
```

```

nodeSelector: {} ①
resources: ②
  limits:
    cpu: 500m
    memory: 256Mi
  requests:
    cpu: 200m
    memory: 128Mi
tolerations: [] ③
# ...

```

1

선택 사항: **nodeSelector** 스텐자는 Pod가 예약된 노드를 정의합니다.

2

resources 스텨자는 **LogFileMetricExporter CR**에 대한 리소스 요구 사항을 정의합니다.

3

선택 사항: **tolerations** 스텨자는 Pod에서 허용하는 허용 오차를 정의합니다.

2.

다음 명령을 실행하여 **LogFileMetricExporter CR**을 적용합니다.

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

검증

logfilesmetricexporter Pod는 각 노드에서 수집기 Pod를 사용하여 동시에 실행됩니다.

•

다음 명령을 실행하고 출력을 관찰하여 **LogFileMetricExporter CR**을 생성한 네임스페이스에서 **logfilesmetricexporter Pod**가 실행되고 있는지 확인합니다.

```
$ oc get pods -l app.kubernetes.io/component=logfilesmetricexporter -n openshift-logging
```

출력 예

NAME	READY	STATUS	RESTARTS	AGE
logfilesmetricexporter-9qbjj	1/1	Running	0	2m46s
logfilesmetricexporter-cbc4v	1/1	Running	0	2m46s

9.5.3. 로그 수집기 CPU 및 메모리 제한 구성

로그 수집기는 CPU 및 메모리 제한을 모두 조정할 수 있습니다.

절차

- 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:
    type: fluentd
  resources:
    limits: ①
      memory: 736Mi
    requests:
      cpu: 100m
      memory: 736Mi
# ...
```

①

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

9.5.4. 입력 수신자 구성

Red Hat OpenShift Logging Operator는 구성된 각 입력 수신자에 대한 서비스를 배포하여 클라이언트가 컬렉터에 쓸 수 있도록 합니다. 이 서비스는 입력 수신자에 지정된 포트를 노출합니다. 서비스 이름

은 다음을 기반으로 생성됩니다.

- 다중 로그 전달자 **ClusterLogForwarder CR** 배포의 경우 서비스 이름은 <**ClusterLogForwarder_CR_name>-<input_name>**> 형식으로 되어 있습니다. 예: **example-http-receiver**.
- 레거시 **ClusterLogForwarder CR** 배포의 경우 이름이 **instance**이고 **openshift-logging** 네임스페이스에 있는 서비스 이름은 **collector-<input_name>** 형식으로 되어 있습니다. 예: **collector-http-receiver**.

9.5.4.1. 감사 로그를 HTTP 서버로 수신하도록 수집기 구성

ClusterLogForwarder 사용자 정의 리소스(CR)에서 **http**를 수신자 입력으로 지정하여 **HTTP** 연결을 수신하고 감사 로그를 **HTTP** 서버로 수신하도록 로그 수집기를 구성할 수 있습니다. 이를 통해 **OpenShift Container Platform** 클러스터 내부 및 외부에서 수집된 감사 로그에 공통 로그 저장소를 사용할 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- **OpenShift CLI(oc)**가 설치되어 있습니다.
- **Red Hat OpenShift Logging Operator**가 설치되어 있습니다.
- **ClusterLogForwarder CR**을 생성했습니다.

절차

1. **http** 수신자 입력 구성을 추가하도록 **ClusterLogForwarder CR**을 수정합니다.

다중 로그 전달자 배포를 사용하는 경우 **ClusterLogForwarder CR**의 예

```

apiVersion: logging.openshift.io/v1beta1
kind: ClusterLogForwarder
metadata:
# ...
spec:
  serviceAccountName: <service_account_name>
  inputs:
    - name: http-receiver ①
      receiver:
        type: http ②
        http:
          format: kubeAPIAudit ③
          port: 8443 ④
  pipelines: ⑤
    - name: http-pipeline
      inputRefs:
        - http-receiver
# ...

```

①

입력 수신자의 이름을 지정합니다.

②

입력 수신자 유형을 **http**로 지정합니다.

③

현재는 **http** 입력 수신자에 대해 **kube-apiserver** 웹 후크 형식만 지원됩니다.

④

선택 사항: 입력 수신자가 수신 대기하는 포트를 지정합니다. 1024 ~65535 사이의 값이어야 합니다. 이 값이 지정되지 않은 경우 기본값은 8443입니다.

⑤

입력 수신자에 대한 파이프라인을 구성합니다.

레거시 배포를 사용하는 경우 **ClusterLogForwarder CR**의 예

```

apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: instance
  namespace: openshift-logging
spec:
  inputs:
    - name: http-receiver ①
      receiver:
        type: http ②
        http:
          format: kubeAPIAudit ③
          port: 8443 ④
  pipelines: ⑤
    - inputRefs:
      - http-receiver
      name: http-pipeline
# ...

```

①

입력 수신자의 이름을 지정합니다.

②

입력 수신자 유형을 **http**로 지정합니다.

③

현재는 **http** 입력 수신자에 대해 **kube-apiserver** 웹 후크 형식만 지원됩니다.

④

선택 사항: 입력 수신자가 수신 대기하는 포트를 지정합니다. 1024 ~65535 사이의 값이어야 합니다. 이 값이 지정되지 않은 경우 기본값은 8443입니다.

⑤

입력 수신자에 대한 파이프라인을 구성합니다.

2.

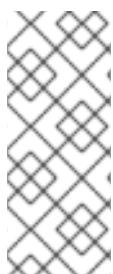
다음 명령을 실행하여 **ClusterLogForwarder CR**에 변경 사항을 적용합니다.

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

추가 리소스

- [API 감사 필터 개요](#)

9.5.5. Fluentd 로그 전달자를 위한 고급 구성



참고

Fluentd는 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다. **Red Hat**은 현재 릴리스 라이프사이클 동안 이 기능에 대한 버그 수정 및 지원을 제공하지만 이 기능은 더 이상 개선 사항을 받지 않습니다. **Fluentd** 대신 **Vector**를 사용할 수 있습니다.

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

- 청크 및 청크 버퍼 크기
- 청크 플러시 동작
- 청크 전달 재시도 동작

Fluentd는 청크라는 단일 **blob**에서 로그 데이터를 수집합니다. **Fluentd**가 청크를 생성할 때 청크는 스테이지에 있는 것으로 간주되어 청크가 데이터로 채워집니다. 청크가 가득 차면 **Fluentd**는 청크를 큐로 이동합니다. 여기서 청크는 플러시되기 전에 보관되거나 대상에 기록됩니다. **Fluentd**는 네트워크 문제 또는 대상의 용량 문제와 같은 여러 가지 이유로 청크를 플러시하지 못할 수 있습니다. 청크를 플러시할 수 없는 경우 **Fluentd**는 구성된 대로 플러시를 다시 시도합니다.

기본적으로 **OpenShift Container Platform**에서 **Fluentd**는 지수 백오프 방법을 사용하여 플러시를 다시 시도합니다. 여기서 **Fluentd**는 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.

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 9.10. 고급 Fluentd 구성 매개변수

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

매개변수	설명	기본
totalLimitSize	스테이지와 큐의 총 크기인 버퍼의 최대 크기입니다. 버퍼 크기가 이 값 을 초과하면 Fluentd는 청크로의 데이터 추가를 중지하고 오류와 함께 실패합니다. 청크에 없는 모든 데이터는 손실됩니다.	모든 출력에 분산된 노드 디스크의 약 15%입니다.
flushInterval	청크 플러시 간격입니다. s (초), m (분), h (시간) 또는 d (일)를 사용 할 수 있습니다.	1s
flushMode	플러시를 수행하는 방법:	간격
	<ul style="list-style-type: none"> ● lazy: timekey: 매개변수를 기반으로 청크를 플러시합니다. timekey 매개변수는 수정할 수 없습니다. ● interval: flushInterval: 매개변수를 기반으로 청크를 플러시 합니다. ● immediate: 데이터가 청크에 추가된 직후 청크를 플러시 합니다. 	
flushThreadCount	청크 플러시를 수행하는 스레드 수입니다. 스레드 수를 늘리면 플러시 처리량이 향상되어 네트워크 대기 시간이 줄어집니다.	2
overflowAction	큐가 가득 찼을 때 청크 동작:	블록
	<ul style="list-style-type: none"> ● throw_exception: 로그에 표시 할 예외를 발생시킵니다. ● block: 전체 버퍼 문제가 해결될 때까지 데이터 청크를 중지합니다. ● drop_oldest_chunk: 새로 들어오는 청크를 수락하기 위해 가장 오래된 청크를 삭제합니다. 오래된 청크는 새로운 청크보다 가치가 적습니다. 	
retryMaxInterval	exponential_backoff 재시도 방법의 최대 시간(초)입니다.	300s

매개변수	설명	기본
retryType	플러시 실패 시 재시도 방법:	exponential_backoff
	<ul style="list-style-type: none"> ● exponential_backoff: 플러시 재시도 사이의 시간을 늘립니다. Fluentd는 retry_max_interval 매개변수에 도달할 때까지 다음 재시도까지 대기하는 시간을 두 배로 늘립니다. ● periodic:retryWait 매개변수를 기반으로 주기적으로 플러시를 재시도 합니다. 	
retryTimeOut	레코드가 삭제되기 전에 재시도 할 최대 시간 간격입니다.	60m
retryWait	다음 청크 플러시 전의 시간(초)입니다.	1s

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

절차

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:
  collection:
    fluentd:
      buffer:
        chunkLimitSize: 8m 1
        flushInterval: 5s 2
        flushMode: interval 3
```

```

flushThreadCount: 3 ④
overflowAction: throw_exception ⑤
retryMaxInterval: "300s" ⑥
retryType: periodic ⑦
retryWait: 1s ⑧
totalLimitSize: 32m ⑨
# ...

```

1

플러시를 위해 큐에 추가되기 전에 각 청크의 최대 크기를 지정합니다.

2

청크 플러시 간격을 지정합니다.

3

`lazy`, `interval` 또는 `immediate` 등 청크 플러시를 수행할 방법을 지정합니다.

4

청크 플러시에 사용할 스레드 수를 지정합니다.

5

`throw_exception`, `block` 또는 `drop_oldest_chunk` 등 큐가 가득 찼을 때의 청크 동작을 지정합니다.

6

`exponential_backoff` 청크 플러시 방법의 최대 간격(초)을 지정합니다.

7

청크 플러시 실패 시 재시도 유형을 `exponential_backoff` 또는 `periodic`으로 지정합니다.

8

다음 청크 플러시 전 시간(초)을 지정합니다.

9

첨크 버퍼의 최대 크기를 지정합니다.

3.

Fluentd Pod가 재배포되었는지 확인합니다.

```
$ oc get pods -l component=collector -n openshift-logging
```

4.

새 값이 fluentd 구성 맵에 있는지 확인합니다.

```
$ oc extract configmap/collector-config --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 "#{ENV['TOTAL_LIMIT_SIZE_PER_BUFFER'] || '8589934592'}"
chunk_limit_size 8m
overflow_action throw_exception
disable_chunk_backup true
</buffer>
```

9.6. 쿠버네티스 이벤트 수집 및 저장

OpenShift Container Platform 이벤트 라우터는 Kubernetes 이벤트를 감시하고 로깅을 통해 수집을 위해 로그를 기록하는 Pod입니다. 이벤트 라우터를 수동으로 배포해야 합니다.

이벤트 라우터는 모든 프로젝트에서 이벤트를 수집하여 STDOUT에 씁니다. 그런 다음 수집기는 이러한 이벤트를 ClusterLogForwarder CR(사용자 정의 리소스)에 정의된 저장소로 전달합니다.

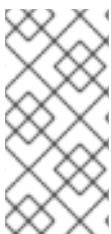


중요

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

9.6.1. 이벤트 라우터 배포 및 구성

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



참고

이벤트 라우터 이미지는 **Red Hat OpenShift Logging Operator**의 일부가 아니며 별도로 다운로드해야 합니다.

다음 템플릿 오브젝트는 이벤트 라우터에 필요한 서비스 계정, 클러스터 역할 및 클러스터 역할 바인딩을 생성합니다. 템플릿은 또한 이벤트 라우터 Pod를 구성하고 배포합니다. 템플릿을 변경하지 않고 사용하거나 템플릿을 편집하여 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.
- Red Hat OpenShift Logging Operator**가 설치되어 있어야 합니다.

절차

- 이벤트 라우터용 템플릿을 생성합니다.

```
apiVersion: template.openshift.io/v1
kind: Template
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
      securityContext:
        allowPrivilegeEscalation: false
        capabilities:
          drop: ["ALL"]
      securityContext:
        runAsNonRoot: true
        seccompProfile:
          type: RuntimeDefault
      volumes:
        - name: config-volume
          configMap:
            name: eventrouter
  parameters:
    - name: IMAGE 6
      displayName: Image
      value: "registry.redhat.io/openshift-logging/eventrouter-rhel8:v0.4"
    - 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 **ClusterRoleBinding**을 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

v0.4 와 같은 태그로 식별되는 이미지를 지정합니다.

7

이벤트 라우터 Pod에 할당할 최소 CPU 양을 지정합니다. 기본값은 100m입니다.

8

이벤트 라우터 Pod에 할당할 최소 메모리 양을 지정합니다. 기본값은 128Mi입니다.

9

오브젝트를 설치할 **openshift-logging** 프로젝트를 지정합니다.

2.

다음 명령을 사용하여 템플릿을 처리하고 적용합니다.

```
$ oc process -f <templatefile> | oc apply -n openshift-logging -
```

예를 들면 다음과 같습니다.

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

출력 예

```
serviceaccount/eventrouter created
clusterrole.rbac.authorization.k8s.io/event-reader created
clusterrolebinding.rbac.authorization.k8s.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.1632d931e88fc8f","namespace":"openshift-service-catalog-removed","selfLink":"/api/v1/namespaces/openshift-service-catalog-removed/events/openshift-service-catalog-controller-manager-remover.1632d931e88fc8f","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 Kibana를 사용할 수도 있습니다.

10장. 로그 스토리지

10.1. 로그 스토리지 정보

로그를 저장하기 위해 클러스터에서 내부 **Loki** 또는 **Elasticsearch** 로그 저장소를 사용하거나 **ClusterLogForwarder** 사용자 정의 리소스(**CR**)를 사용하여 로그를 외부 저장소로 전달할 수 있습니다.

10.1.1. 로그 스토리지 유형

Loki는 수평으로 확장 가능한 고가용성 다중 테넌트 로그 수집 시스템으로, 로깅을 위한 로그 저장소로 **Elasticsearch**의 대안으로 제공됩니다.

Elasticsearch는 들어오는 로그 레코드를 수집 중에 완전히 인덱싱합니다. **Loki**는 수집 중에 몇 가지 고정된 레이블만 인덱싱하고 로그를 저장한 후 더 복잡한 구문 분석을 수행합니다. 즉 **Loki**는 로그를 더 빠르게 수집할 수 있습니다.

10.1.1.1. Elasticsearch 로그 저장소 정보

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

Elasticsearch는 **Fluentd**의 로그 데이터를 데이터 저장소 또는 인덱스로 구성한 다음 각 인덱스를 **shards**라고 하는 조각 여러 개로 다시 세분화합니다. 그리고 이 조각을 **Elasticsearch** 클러스터의 **Elasticsearch** 노드 세트에 분산 배치합니다. 복제본이라는 이름의 **shard** 사본을 작성하도록 **Elasticsearch**를 구성할 수 있습니다. **Elasticsearch**는 이 역시 **Elasticsearch** 노드에 분산 배치합니다. **ClusterLogging** 사용자 정의 리소스(**CR**)를 사용하면 **shard**의 복제 방식을 지정하여 데이터 중복성과 장애에 대한 회복 탄력성을 제공할 수 있습니다. **ClusterLogging CR**의 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 Operator**는 각 **Elasticsearch** 노드가 자체 스토리지 볼륨이 있는 고유한 배포를 사용하여 배포되도록 합니다. 필요에 따

라 **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.

10.1.2. 로그 저장소 쿼리

[LogQL](#) 로그 쿼리 언어를 사용하여 **Loki**를 쿼리 할 수 있습니다.

10.1.3. 추가 리소스

- [Loki 구성 요소 문서](#)
- [Loki 오브젝트 스토리지 문서](#)

10.2. 로그 스토리지 설치

[OpenShift CLI\(oc\)](#) 또는 [OpenShift Container Platform](#) 웹 콘솔을 사용하여 [OpenShift Container Platform](#) 클러스터에 로그 저장소를 배포할 수 있습니다.



참고

로깅 5.9 릴리스에는 업데이트된 **OpenShift Elasticsearch Operator** 버전이 포함되어 있지 않습니다. 현재 **Logging 5.8**과 함께 릴리스된 **OpenShift Elasticsearch Operator**를 사용하는 경우 로깅 5.8의 EOL까지 로깅에서 계속 작동합니다. **OpenShift Elasticsearch Operator**를 사용하여 기본 로그 스토리지를 관리하는 대신 **Loki Operator**를 사용할 수 있습니다. 로깅 라이프사이클 날짜에 대한 자세한 내용은 [Platform Agnostic Operators](#) 를 참조하십시오.

10.2.1. Loki 로그 저장소 배포

Loki Operator를 사용하여 **OpenShift Container Platform** 클러스터에 내부 **Loki** 로그 저장소를 배포 할 수 있습니다. **Loki Operator**를 설치한 후 보안을 생성하여 **Loki** 오브젝트 스토리지를 구성하고 **LokiStack CR**(사용자 정의 리소스)을 생성해야 합니다.

10.2.1.1. Loki 배포 크기 조정

Loki의 크기 조정은 **1x.<size>** 형식을 따릅니다. 여기서 **1x** 값은 인스턴스 수이고 **<size>**는 성능 기능을 지정합니다.

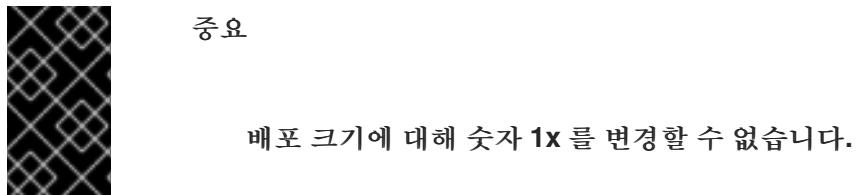


표 10.1. Loki 크기 조정

	1x.demo	1x.extra-small	1x.small	1x.medium
데이터 전송	데모만 사용	100GB/일	500GB/day	2TB/day
초당 쿼리(QPS)	데모만 사용	200ms에서 1-25 QPS	25-50 QPS (200ms)	25-75 QPS (200ms)
복제 요인	없음	2	2	2
총 CPU 요청	없음	14개의 vCPU	34 vCPU	54 vCPU
ruler을 사용하는 경우 총 CPU 요청	없음	16개의 vCPU	42 vCPU	70개의 vCPU
총 메모리 요청	없음	31Gi	67Gi	139Gi
룰러를 사용하는 경우 총 메모리 요청	없음	35Gi	83Gi	171Gi
총 디스크 요청	40Gi	430Gi	430Gi	590Gi
ruler을 사용하는 경우 총 디스크 요청	80Gi	750Gi	750Gi	910Gi

10.2.1.2. OpenShift Container Platform 웹 콘솔을 사용하여 Loki Operator 설치

OpenShift Container Platform 클러스터에 로깅을 설치하고 구성하려면 추가 **Operator**를 설치해야 합니다. 이 작업은 웹 콘솔 내의 **Operator Hub**에서 수행할 수 있습니다.

OpenShift Container Platform Operator는 CR(사용자 정의 리소스)을 사용하여 애플리케이션 및 해당 구성 요소를 관리합니다. 높은 수준의 구성 및 설정은 CR 내에서 사용자가 제공합니다. **Operator**는 고급 지시문을 **Operator** 논리에 포함된 모범 사례를 기반으로 하위 수준 작업으로 변환합니다. CRD(사용자 정의 리소스 정의)는 CR을 정의하고 **Operator** 사용자가 사용할 수 있는 모든 구성은 나열합니다. **Operator**를 설치하면 CRD가 생성되는 CR을 생성하는 데 사용됩니다.

사전 요구 사항

- 지원되는 오브젝트 저장소(AWS S3, Google Cloud Storage, Azure, Swift, Minio, OpenShift Data Foundation)에 액세스할 수 있습니다.
- 관리자 권한이 있습니다.
- **OpenShift Container Platform** 웹 콘솔에 액세스할 수 있습니다.

절차

1. **OpenShift Container Platform** 웹 콘솔 관리자 화면에서 **Operator** → **OperatorHub**로 이동합니다.
2. 키워드로 필터링 필드에 **Loki Operator**를 입력합니다. 사용 가능한 **Operator** 목록에서 **Loki Operator**를 클릭한 다음 설치를 클릭합니다.



중요

Community Loki Operator는 Red Hat에서 지원하지 않습니다.

3. 업데이트 채널로 **stable** 또는 **stable-x.y**를 선택합니다.



참고

stable 채널은 최신 로깅 릴리스에 대한 업데이트만 제공합니다. 이전 릴리스에 대한 업데이트를 계속 받으려면 서브스크립션 채널을 **stable-x.y**로 변경해야 합니다. 여기서 **x.y**는 설치한 로깅 및 마이너 버전을 나타냅니다. 예를 들면 **stable-5.7**입니다.

Loki Operator는 글로벌 **Operator** 그룹 네임스페이스 **openshift-operators-redhat**에 배포되어야 하므로 설치 모드 및 설치된 네임스페이스가 이미 선택되어 있습니다. 이 네임스페이스가 아직 없는 경우 이를 위해 생성됩니다.

4.

이 네임스페이스에서 **operator-recommended** 클러스터 모니터링 사용을 선택합니다.

이 옵션은 **Namespace** 오브젝트에서 **openshift.io/cluster-monitoring: "true"** 라벨을 설정합니다. 클러스터 모니터링이 **openshift-operators-redhat** 네임스페이스를 스크랩하도록 하려면 이 옵션을 선택해야 합니다.

5.

업데이트 승인의 경우 자동을 선택한 다음 설치를 클릭합니다.

서브스크립션의 승인 전략이 자동으로 설정된 경우 선택한 채널에서 새 **Operator** 버전을 사용할 수 있는 즉시 업데이트 프로세스가 시작됩니다. 승인 전략이 **Manual**로 설정된 경우 보류 중인 업데이트를 수동으로 승인해야 합니다.

검증

1.

Operator → 설치된 **Operator**로 이동합니다.

2.

openshift-logging 프로젝트가 선택되어 있는지 확인합니다.

3.

상태 열에서 **InstallSucceeded** 가 포함된 녹색 확인 표시와 최대 날짜 텍스트가 표시되는지 확인합니다.



참고

Operator는 설치가 완료되기 전에 실패 상태를 표시할 수 있습니다. **Operator** 설치가 **InstallSucceeded** 메시지와 함께 완료되면 페이지를 새로 고칩니다.

10.2.1.3. 웹 콘솔을 사용하여 **Loki** 오브젝트 스토리지의 보안 생성

Loki 오브젝트 스토리지를 구성하려면 보안을 생성해야 합니다. **OpenShift Container Platform** 웹 콘솔을 사용하여 시크릿을 생성할 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- **OpenShift Container Platform** 웹 콘솔에 액세스할 수 있습니다.
- **Loki Operator**를 설치했습니다.

절차

1. **OpenShift Container Platform** 웹 콘솔의 관리자 관점에서 워크로드 → 시크릿으로 이동합니다.
2. 생성 드롭다운 목록에서 **YAML**에서 선택합니다.
3. **access_key_id** 및 **access_key_secret** 필드를 사용하여 인증 정보 및 버킷 이름, 끝점, 지역 필드를 지정하여 오브젝트 스토리지 위치를 정의하는 시크릿을 생성합니다. **AWS**는 다음 예제에서 사용됩니다.

Secret 오브젝트의 예

```
apiVersion: v1
kind: Secret
metadata:
```

```

name: logging-loki-s3
namespace: openshift-logging
stringData:
  access_key_id: AKIAIOSFODNN7EXAMPLE
  access_key_secret: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
  bucketnames: s3-bucket-name
  endpoint: https://s3.eu-central-1.amazonaws.com
  region: eu-central-1

```

추가 리소스

- [Loki 오브젝트 스토리지](#)

10.2.1.4. 웹 콘솔을 사용하여 **LokiStack** 사용자 정의 리소스 생성

OpenShift Container Platform 웹 콘솔을 사용하여 **LokiStack CR**(사용자 정의 리소스)을 생성할 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- **OpenShift Container Platform** 웹 콘솔에 액세스할 수 있습니다.
- **Loki Operator**를 설치했습니다.

절차

1. **Operator** → 설치된 **Operator** 페이지로 이동합니다. 모든 인스턴스 탭을 클릭합니다.
2. **Create new** 드롭다운 목록에서 **LokiStack** 을 선택합니다.
3. **YAML** 보기와 선택한 다음 다음 템플릿을 사용하여 **LokiStack CR**을 생성합니다.

```

apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki ①
  namespace: openshift-logging
spec:
  size: 1x.small ②
  storage:
    schemas:
      - version: v12
        effectiveDate: '2022-06-01'
    secret:
      name: logging-loki-s3 ③
      type: s3 ④
  storageClassName: <storage_class_name> ⑤
  tenants:
    mode: openshift-logging ⑥

```

①

logging-loki라는 이름을 사용합니다.

②

배포 크기를 지정합니다. 로깅 5.8 이상 버전에서 **Loki**의 프로덕션 인스턴스에 지원되는 크기 옵션은 **1x.extra- #159**, **1x.windows** 또는 **1x.medium**입니다.



중요

배포 크기에 대해 숫자 **1x**를 변경할 수 없습니다.

③

로그 스토리지에 사용되는 시크릿을 지정합니다.

④

해당 스토리지 유형을 지정합니다.

⑤

임시 스토리지의 스토리지 클래스 이름을 입력합니다. 최상의 성능을 위해서는 블록 스토리지를 할당하는 스토리지 클래스를 지정합니다. **oc get storageclasses** 명령을 사용하여 클러스터에 사용 가능한 스토리지 클래스를 나열할 수 있습니다.

6

LokiStack은 기본적으로 다중 테넌트 모드에서 실행되며 수정할 수 없습니다. 각 로그 유형(audit, infrastructure, application logs)에 대해 하나의 테넌트가 제공됩니다. 이를 통해 개별 사용자 및 사용자 그룹에 대한 액세스 제어를 다른 로그 스트림에 사용할 수 있습니다.

4. 생성을 클릭합니다.

10.2.1.5. CLI를 사용하여 Loki Operator 설치

OpenShift Container Platform 클러스터에 로깅을 설치하고 구성하려면 추가 **Operator**를 설치해야 합니다. 이 작업은 **OpenShift Container Platform CLI**에서 수행할 수 있습니다.

OpenShift Container Platform Operator는 CR(사용자 정의 리소스)을 사용하여 애플리케이션 및 해당 구성 요소를 관리합니다. 높은 수준의 구성 및 설정은 CR 내에서 사용자가 제공합니다. **Operator**는 고급 지시문을 **Operator** 논리에 포함된 모범 사례를 기반으로 하위 수준 작업으로 변환합니다. **CRD**(사용자 정의 리소스 정의)는 CR을 정의하고 **Operator** 사용자가 사용할 수 있는 모든 구성은 나열합니다. **Operator**를 설치하면 CRD가 생성되는 CR을 생성하는 데 사용됩니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- **OpenShift CLI(oc)**를 설치합니다.
- 지원되는 오브젝트 저장소에 액세스할 수 있습니다. 예를 들어 **AWS S3, Google Cloud Storage, Azure, Swift, Minio** 또는 **OpenShift Data Foundation**입니다.

절차

1. **Subscription** 오브젝트를 생성합니다.

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: loki-operator
```

```

namespace: openshift-operators-redhat ①
spec:
  channel: stable ②
  name: loki-operator
  source: redhat-operators ③
  sourceNamespace: openshift-marketplace

```

1

openshift-operators-redhat 네임스페이스를 지정해야 합니다.

2

stable 또는 **stable -5.<y>**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.

2.

Subscription 오브젝트를 적용합니다.

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

10.2.1.6. CLI를 사용하여 Loki 오브젝트 스토리지의 보안 생성

Loki 오브젝트 스토리지를 구성하려면 보안을 생성해야 합니다. OpenShift CLI(**oc**)를 사용하여 이 작업을 수행할 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- Loki Operator를 설치했습니다.
- OpenShift CLI(**oc**)를 설치했습니다.

절차

- 다음 명령을 실행하여 인증서 및 키 파일이 포함된 디렉터리에 보안을 생성합니다.

```
$ oc create secret generic -n openshift-logging <your_secret_name> \
--from-file=tls.key=<your_key_file>
--from-file=tls.crt=<your_crt_file>
--from-file=ca-bundle.crt=<your_bundle_file>
--from-literal=username=<your_username>
--from-literal=password=<your_password>
```



참고

최상의 결과를 위해 일반 또는 불투명한 보안을 사용하십시오.

검증

- 다음 명령을 실행하여 보안이 생성되었는지 확인합니다.

```
$ oc get secrets
```

추가 리소스

- [Loki 오브젝트 스토리지](#)

10.2.1.7. CLI를 사용하여 LokiStack 사용자 정의 리소스 생성

OpenShift CLI(oc)를 사용하여 LokiStack CR(사용자 정의 리소스)을 생성할 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- [Loki Operator](#)를 설치했습니다.
- [OpenShift CLI\(oc\)](#)를 설치합니다.

절차

1.

LokiStack CR을 생성합니다.

LokiStack CR의 예

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
  size: 1x.small ①
  storage:
    schemas:
      - version: v12
        effectiveDate: "2022-06-01"
    secret:
      name: logging-loki-s3 ②
      type: s3 ③
  storageClassName: <storage_class_name> ④
  tenants:
    mode: openshift-logging ⑤
```

1

배포 크기를 지정합니다. 로깅 5.8 이상 버전에서 Loki의 프로덕션 인스턴스에 지원되는 크기 옵션은 **1x.extra-#159**, **1x.windows** 또는 **1x.medium**입니다.



중요

배포 크기에 대해 숫자 **1x**를 변경할 수 없습니다.

2

로그 저장소 시크릿의 이름을 지정합니다.

3

로그 저장소 시크릿 유형을 지정합니다.

4

임시 스토리지의 스토리지 클래스 이름을 지정합니다. 최상의 성능을 위해서는 블록 스토리지를 할당하는 스토리지 클래스를 지정합니다. **oc get storageclasses** 명령을 사용하여 클러스터에 사용 가능한 스토리지 클래스를 나열할 수 있습니다.

5

LokiStack은 기본적으로 다중 테넌트 모드에서 실행되며 수정할 수 없습니다. 각 로그 유형(**audit**, **infrastructure**, **application logs**)에 대해 하나의 테넌트가 제공됩니다. 이를 통해 개별 사용자 및 사용자 그룹에 대한 액세스 제어를 다른 로그 스트림에 사용할 수 있습니다.

2.

다음 명령을 실행하여 **LokiStack CR**을 적용합니다.

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

검증

-

다음 명령을 실행하고 출력을 관찰하여 **openshift-logging** 프로젝트에 **Pod**를 나열하여 설치를 확인합니다.

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

다음 목록과 유사하게 로깅 구성 요소에 대한 여러 **Pod**가 표시되는지 확인합니다.

출력 예

NAME	READY	STATUS	RESTARTS	AGE
cluster-logging-operator-78fddc697-mnl82	1/1	Running	0	14m
collector-6cgqlq	2/2	Running	0	45s
collector-8r664	2/2	Running	0	45s
collector-8z7px	2/2	Running	0	45s
collector-pdxi9	2/2	Running	0	45s
collector-tc9dx	2/2	Running	0	45s
collector-xkd76	2/2	Running	0	45s
logging-loki-compactor-0	1/1	Running	0	8m2s
logging-loki-distributor-b85b7d9fd-25j9g	1/1	Running	0	8m2s

```

logging-loki-distributor-b85b7d9fd-xwjs6      1/1  Running  0   8m2s
logging-loki-gateway-7bb86fd855-hjhl4        2/2  Running  0   8m2s
logging-loki-gateway-7bb86fd855-qjtlb        2/2  Running  0   8m2s
logging-loki-index-gateway-0                 1/1  Running  0   8m2s
logging-loki-index-gateway-1                 1/1  Running  0   7m29s
logging-loki-ingester-0                      1/1  Running  0   8m2s
logging-loki-ingester-1                      1/1  Running  0   6m46s
logging-loki-querier-f5cf9cb87-9fdjd       1/1  Running  0   8m2s
logging-loki-querier-f5cf9cb87-fp9v5        1/1  Running  0   8m2s
logging-loki-query-frontend-58c579fcb7-lfvbc 1/1  Running  0   8m2s
logging-loki-query-frontend-58c579fcb7-tjf9k 1/1  Running  0   8m2s
logging-view-plugin-79448d8df6-ckgmx        1/1  Running  0   46s

```

10.2.2. Loki 오브젝트 스토리지

Loki Operator는 AWS S3 및 Minio 및 OpenShift Data Foundation과 같은 기타 S3 호환 오브젝트 저장소를 지원합니다. Azure, GCS 및 Swift도 지원됩니다.

Loki 스토리지에 권장되는 nomenclature는 `logging-loki- <your_storage_provider>`입니다.

다음 표는 각 스토리지 공급자에 대한 LokiStack CR(사용자 정의 리소스) 내의 유형 값을 보여줍니다. 자세한 내용은 스토리지 공급자의 섹션을 참조하십시오.

표 10.2. 시크릿 유형 빠른 참조

스토리지 공급자	보안 유형 값
AWS	s3
Azure	azure
Google Cloud	gcs
Minio	s3
OpenShift Data Foundation	s3
Swift	swift

10.2.2.1. AWS 스토리지

사전 요구 사항

- **Loki Operator**를 설치했습니다.
- **OpenShift CLI(oc)**를 설치합니다.
- **AWS**에 버킷 을 생성했습니다.
- **AWS IAM 정책 및 IAM 사용자를** 생성했습니다.

절차

- 다음 명령을 실행하여 이름 **logging-loki-aws** 를 사용하여 오브젝트 스토리지 시크릿을 생성합니다.

```
$ oc create secret generic logging-loki-aws \
--from-literal=bucketnames=<bucket_name> \
--from-literal=endpoint=<aws_bucket_endpoint> \
--from-literal=access_key_id=<aws_access_key_id> \
--from-literal=access_key_secret=<aws_access_key_secret> \
--from-literal=region=<aws_region_of_your_bucket>"
```

10.2.2.2. Azure 스토리지

사전 요구 사항

- **Loki Operator**를 설치했습니다.
- **OpenShift CLI(oc)**를 설치합니다.
- **Azure**에 버킷 을 생성했습니다.

절차

- 다음 명령을 실행하여 **name logging-loki-azure** 를 사용하여 오브젝트 스토리지 시크릿을 생성합니다.

```
$ oc create secret generic logging-loki-azure \
--from-literal=container=<azure_container_name>" \
--from-literal=environment=<azure_environment>" \ ①
--from-literal=account_name=<azure_account_name>" \
--from-literal=account_key=<azure_account_key>"
```

①

지원되는 환경 값은 **AzureGlobal**, **AzureChinaCloud**, **AzureGermanCloud** 또는 **AzureUSGovernment** 입니다.

10.2.2.3. Google Cloud Platform 스토리지

사전 요구 사항

- **Loki Operator**를 설치했습니다.
- **OpenShift CLI(oc)**를 설치합니다.
- **GCP(Google Cloud Platform)**에 [프로젝트](#)를 생성했습니다.
- 동일한 프로젝트에 [버킷](#)을 생성했습니다.
- **GCP** 인증을 위해 동일한 프로젝트에 [서비스 계정](#)을 생성하셨습니다.

절차

1. **GCP**에서 수신한 서비스 계정 인증 정보를 **key.json**이라는 파일에 복사합니다.
2. 다음 명령을 실행하여 이름 **logging-loki-gcs**를 사용하여 오브젝트 스토리지 시크릿을 생성합니다.

```
$ oc create secret generic logging-loki-gcs \
--from-literal=bucketname=<bucket_name>" \
--from-file=key.json=<path/to/key.json>"
```

10.2.2.4. Minio 스토리지

사전 요구 사항

- **Loki Operator**를 설치했습니다.
- **OpenShift CLI(oc)**를 설치합니다.
- **Minio** 가 클러스터에 배포되어 있습니다.
- **Minio**에 버킷을 생성했습니다.

절차

- 다음 명령을 실행하여 **name logging-loki-minio** 를 사용하여 오브젝트 스토리지 시크릿을 생성합니다.

```
$ oc create secret generic logging-loki-minio \
--from-literal=bucketnames=<bucket_name> \
--from-literal=endpoint=<minio_bucket_endpoint> \
--from-literal=access_key_id=<minio_access_key_id> \
--from-literal=access_key_secret=<minio_access_key_secret>"
```

10.2.2.5. OpenShift Data Foundation 스토리지

사전 요구 사항

- **Loki Operator**를 설치했습니다.
- **OpenShift CLI(oc)**를 설치합니다.
- **OpenShift Data Foundation** 을 배포했습니다.

-

[오브젝트 스토리지를 위해 OpenShift Data Foundation 클러스터를 구성했습니다.](#)

절차

1.

openshift-logging 네임스페이스에 **ObjectBucketClaim** 사용자 정의 리소스를 생성합니다.

```
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: loki-bucket-odf
  namespace: openshift-logging
spec:
  generateBucketName: loki-bucket-odf
  storageClassName: openshift-storage.noobaa.io
```

2.

다음 명령을 실행하여 관련 **ConfigMap** 오브젝트에서 버킷 속성을 가져옵니다.

```
BUCKET_HOST=$(oc get -n openshift-logging configmap loki-bucket-odf -o jsonpath='{.data.BUCKET_HOST}')
BUCKET_NAME=$(oc get -n openshift-logging configmap loki-bucket-odf -o jsonpath='{.data.BUCKET_NAME}')
BUCKET_PORT=$(oc get -n openshift-logging configmap loki-bucket-odf -o jsonpath='{.data.BUCKET_PORT}')
```

3.

다음 명령을 실행하여 관련 시크릿에서 버킷 액세스 키를 가져옵니다.

```
ACCESS_KEY_ID=$(oc get -n openshift-logging secret loki-bucket-odf -o jsonpath='{.data.AWS_ACCESS_KEY_ID}' | base64 -d)
SECRET_ACCESS_KEY=$(oc get -n openshift-logging secret loki-bucket-odf -o jsonpath='{.data.AWS_SECRET_ACCESS_KEY}' | base64 -d)
```

4.

다음 명령을 실행하여 **logging-loki-odf**라는 이름으로 오브젝트 스토리지 시크릿을 생성합니다.

```
$ oc create -n openshift-logging secret generic logging-loki-odf \
--from-literal=access_key_id=<access_key_id> \
--from-literal=access_key_secret=<secret_access_key> \
--from-literal=bucketnames=<bucket_name> \
--from-literal=endpoint="https://<bucket_host>:<bucket_port>"
```

10.2.2.6. Swift 스토리지

사전 요구 사항

- Loki Operator를 설치했습니다.
- OpenShift CLI(oc)를 설치합니다.
- Swift에 버킷 을 생성했습니다.

절차

- 다음 명령을 실행하여 이름 **logging-loki-swift**를 사용하여 오브젝트 스토리지 시크릿을 생성합니다.

```
$ oc create secret generic logging-loki-swift \
--from-literal=auth_url=<swift_auth_url> \
--from-literal=username=<swift_usernameclaim> \
--from-literal=user_domain_name=<swift_user_domain_name> \
--from-literal=user_domain_id=<swift_user_domain_id> \
--from-literal=user_id=<swift_user_id> \
--from-literal=password=<swift_password> \
--from-literal=domain_id=<swift_domain_id> \
--from-literal=domain_name=<swift_domain_name> \
--from-literal=container_name=<swift_container_name>"
```

- 선택적으로 다음 명령을 실행하여 프로젝트별 데이터, 지역 또는 둘 다를 제공할 수 있습니다.

```
$ oc create secret generic logging-loki-swift \
--from-literal=auth_url=<swift_auth_url> \
--from-literal=username=<swift_usernameclaim> \
--from-literal=user_domain_name=<swift_user_domain_name> \
--from-literal=user_domain_id=<swift_user_domain_id> \
--from-literal=user_id=<swift_user_id> \
--from-literal=password=<swift_password> \
--from-literal=domain_id=<swift_domain_id> \
--from-literal=domain_name=<swift_domain_name> \
--from-literal=container_name=<swift_container_name> \
--from-literal=project_id=<swift_project_id> \
--from-literal=project_name=<swift_project_name> \
--from-literal=project_domain_id=<swift_project_domain_id> \
--from-literal=project_domain_name=<swift_project_domain_name> \
--from-literal=region=<swift_region>"
```

10.2.3. Elasticsearch 로그 저장소 배포

OpenShift Elasticsearch Operator를 사용하여 OpenShift Container Platform 클러스터에 내부 Elasticsearch 로그 저장소를 배포할 수 있습니다.

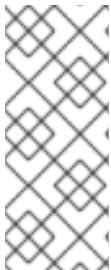


참고

로깅 5.9 릴리스에는 업데이트된 OpenShift Elasticsearch Operator 버전이 포함되어 있지 않습니다. 현재 Logging 5.8과 함께 릴리스된 OpenShift Elasticsearch Operator를 사용하는 경우 로깅 5.8의 EOL까지 로깅에서 계속 작동합니다. OpenShift Elasticsearch Operator를 사용하여 기본 로그 스토리지를 관리하는 대신 Loki Operator를 사용할 수 있습니다. 로깅 라이프사이클 날짜에 대한 자세한 내용은 [Platform Agnostic Operators](#)를 참조하십시오.

10.2.3.1. Elasticsearch의 스토리지 고려 사항

각 Elasticsearch 배포 구성에는 영구 볼륨이 필요합니다. OpenShift Container Platform에서는 PVC(영구 볼륨 클레임)를 사용합니다.



참고

영구 스토리지에 로컬 볼륨을 사용하는 경우 LocalVolume 개체에서 volumeMode: block에 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 저널 및 /var/log/containers/*.log 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. 그

러나 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.

10.2.3.2. 웹 콘솔을 사용하여 OpenShift Elasticsearch Operator 설치

OpenShift Elasticsearch Operator는 OpenShift Logging에 사용되는 Elasticsearch 클러스터를 생성하고 관리합니다.

사전 요구 사항

- Elasticsearch는 메모리를 많이 사용하는 애플리케이션입니다. ClusterLogging 사용자 정의 리소스에서 달리 지정하지 않는 한 각 Elasticsearch 노드에는 메모리 요청 및 제한 모두에 최소 16GB의 메모리가 필요합니다.

초기 OpenShift Container Platform 노드 세트는 Elasticsearch 클러스터를 지원하기에 충분히 크지 않을 수 있습니다. 권장 메모리 이상에서 각 Elasticsearch 노드에 대해 최대 64GB까지 실행하려면 OpenShift Container Platform 클러스터에 노드를 추가해야 합니다.

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

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



참고

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

절차

1. OpenShift Container Platform 웹 콘솔에서 **Operator** → **OperatorHub**를 클릭합니다.
2. 사용 가능한 **Operator** 목록에서 **OpenShift Elasticsearch Operator**를 클릭하고 설치를 클릭합니다.
3. 설치 모드에서 클러스터의 모든 네임스페이스가 선택되어 있는지 확인합니다.
4. 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 있으며, 이로 인해 **OpenShift Container Platform** 지표와 동일한 이름의 지표를 계시할 수 있으므로 충돌이 발생합니다.
5. 이 네임스페이스에서 **Operator** 권장 클러스터 모니터링 사용을 선택합니다.

이 옵션은 **Namespace** 오브젝트에서 **openshift.io/cluster-monitoring: "true"** 라벨을 설정합니다. 클러스터 모니터링이 **openshift-operators-redhat** 네임스페이스를 스크랩하도록 하려면 이 옵션을 선택해야 합니다.
6. **stable-5.x**를 업데이트 채널로 선택합니다.
7. 업데이트 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.
8. 설치를 클릭합니다.

검증

1. **Operator** → 설치된 **Operator** 페이지로 전환하여 **OpenShift Elasticsearch Operator**가 설치되었는지 확인합니다.
2. 상태가 성공인 모든 프로젝트에 **OpenShift Elasticsearch Operator**가 나열되어 있는지 확인합니다.

10.2.3.3. CLI를 사용하여 OpenShift Elasticsearch Operator 설치

OpenShift CLI(oc)를 사용하여 **OpenShift Elasticsearch Operator**를 설치할 수 있습니다.

사전 요구 사항

- **Elasticsearch**에 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. **Elasticsearch**는 원시 블록 볼륨을 사용할 수 없습니다.

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

- 관리자 권한이 있습니다.
- **OpenShift CLI(oc)**가 설치되어 있습니다.

절차

1.

Namespace 오브젝트를 **YAML** 파일로 생성합니다.

```
apiVersion: v1
kind: Namespace
metadata:
  name: openshift-operators-redhat ①
  annotations:
    openshift.io/node-selector: ""
  labels:
    openshift.io/cluster-monitoring: "true" ②
```

①

openshift-operators-redhat 네임스페이스를 지정해야 합니다. 지표와의 충돌을 방지 하려면 **openshift-operators** 네임스페이스가 아닌 **openshift-operators-redhat** 네임스페이스에서 지표를 스크랩하도록 **Prometheus** 클러스터 모니터링 스택을 구성합니다. **openshift-operators** 네임스페이스에 신뢰할 수 없는 커뮤니티 Operator가 포함될 수 있으며 지표와 동일한 이름의 지표를 게시하면 충돌이 발생합니다.

②

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

2.

다음 명령을 실행하여 **Namespace** 오브젝트를 적용합니다.

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

3.

OperatorGroup 오브젝트를 **YAML** 파일로 생성합니다.

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: openshift-operators-redhat
  namespace: openshift-operators-redhat ①
spec: {}
```

①

openshift-operators-redhat 네임스페이스를 지정해야 합니다.

4.

다음 명령을 실행하여 **OperatorGroup** 오브젝트를 적용합니다.

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

5.

OpenShift Elasticsearch Operator에 네임스페이스를 서브스크립션 할 **Subscription** 오브젝트를 생성합니다.

서브스크립션의 예

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: elasticsearch-operator
  namespace: openshift-operators-redhat ①
spec:
  channel: stable-x.y ②
  installPlanApproval: Automatic ③
  source: redhat-operators ④
  sourceNamespace: openshift-marketplace
  name: elasticsearch-operator
```

1

openshift-operators-redhat 네임스페이스를 지정해야 합니다.

2

stable 또는 **stable-x.y** 를 채널로 지정합니다. 다음 참고 사항을 참조하십시오.

3

자동으로 새 버전을 사용할 수 있을 때 **OLM(Operator Lifecycle Manager)**이 **Operator**를 자동으로 업데이트할 수 있습니다. 수동 에서는 적절한 인증 정보를 가진 사용자가 **Operator** 업데이트를 승인해야 합니다.

4

redhat-operators를 지정합니다. **OpenShift Container Platform** 클러스터가 제한된 네트워크(연결이 끊긴 클러스터)에 설치된 경우 **OLM(Operator Lifecycle Manager)**을 구

성할 때 생성된 **CatalogSource** 오브젝트의 이름을 지정합니다.



참고

stable을 지정하면 안정적인 최신 릴리스의 현재 버전이 설치됩니다.
installPlanApproval: "Automatic"과 함께 **stable**을 사용하면 **Operator**가 안정적인 최신 메이저 및 마이너 릴리스로 자동 업그레이드됩니다.

stable-x.y를 지정하면 특정 주요 릴리스의 현재 마이너 버전이 설치됩니다.
installPlanApproval: "Automatic"과 함께 **stable-x.y**를 사용하면 **Operator**가 주요 릴리스 내에서 안정적인 최신 마이너 릴리스로 자동 업그레이드됩니다.

6.

다음 명령을 실행하여 서브스크립션을 적용합니다.

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

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

검증

1.

다음 명령을 실행합니다.

```
$ oc get csv -n --all-namespaces
```

2.

출력을 관찰하고 **OpenShift Elasticsearch Operator**의 Pod가 각 네임스페이스에 있는지 확인합니다.

출력 예

NAMESPACE	VERSION	REPLACES	NAME	DISPLAY
			PHASE	
			elasticsearch-operator.v5.8.1	OpenShift
			elasticsearch-operator.v5.8.0	Succeeded
			elasticsearch-operator.v5.8.1	OpenShift

```

Elasticsearch Operator 5.8.1     .elasticsearch-operator.v5.8.0 Succeeded
kube-public                      .elasticsearch-operator.v5.8.1 OpenShift
Elasticsearch Operator 5.8.1      .elasticsearch-operator.v5.8.0 Succeeded
kube-system                       .elasticsearch-operator.v5.8.1 OpenShift
Elasticsearch Operator 5.8.1      .elasticsearch-operator.v5.8.0 Succeeded
non-destructive-test              .elasticsearch-operator.v5.8.1 OpenShift
Elasticsearch Operator 5.8.1      .elasticsearch-operator.v5.8.0 Succeeded
openshift-apiserver-operator     .elasticsearch-operator.v5.8.1 OpenShift
Elasticsearch Operator 5.8.1      .elasticsearch-operator.v5.8.0 Succeeded
openshift-apiserver               .elasticsearch-operator.v5.8.1 OpenShift
Elasticsearch Operator 5.8.1      .elasticsearch-operator.v5.8.0 Succeeded
...

```

10.2.4. 로그 스토리지 구성

ClusterLogging 사용자 정의 리소스(CR)를 수정하여 로깅에서 사용하는 로그 스토리지 유형을 구성할 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- **OpenShift CLI(oc)**가 설치되어 있습니다.
- **Red Hat OpenShift Logging Operator**와 **LokiStack** 또는 **Elasticsearch**인 내부 로그 저장소를 설치했습니다.
- **ClusterLogging CR**을 생성했습니다.



참고

로깅 5.9 릴리스에는 업데이트된 **OpenShift Elasticsearch Operator** 버전이 포함되어 있지 않습니다. 현재 **Logging 5.8**과 함께 릴리스된 **OpenShift Elasticsearch Operator**를 사용하는 경우 로깅 5.8의 EOL까지 로깅에서 계속 작동합니다. **OpenShift Elasticsearch Operator**를 사용하여 기본 로그 스토리지를 관리하는 대신 **Loki Operator**를 사용할 수 있습니다. 로깅 라이프사이클 날짜에 대한 자세한 내용은 [Platform Agnostic Operators](#)를 참조하십시오.

절차

1.

ClusterLogging CR logStore 사양을 수정합니다.

ClusterLogging CR 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
# ...
spec:
# ...
logStore:
  type: <log_store_type> ①
  elasticsearch: ②
    nodeCount: <integer>
    resources: {}
    storage: {}
  redundancyPolicy: <redundancy_type> ③
  lokistack: ④
    name: {}
# ...
```

1

로그 저장소 유형을 지정합니다. **lokistack** 또는 **elasticsearch** 일 수 있습니다.

2

Elasticsearch 로그 저장소에 대한 선택적 구성 옵션입니다.

3

중복 유형을 지정합니다. 이 값은 **ZeroRedundancy**, **SingleRedundancy**, **MultipleRedundancy** 또는 **FullRedundancy** 일 수 있습니다.

4

LokiStack에 대한 선택적 구성 옵션입니다.

LokiStack을 로그 저장소로 지정하는 ClusterLogging CR의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
  name: instance
  namespace: openshift-logging
spec:
  managementState: Managed
  logStore:
    type: lokistack
    lokistack:
      name: logging-loki
# ...
```

2.

다음 명령을 실행하여 ClusterLogging CR을 적용합니다.

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

10.3. LOKISTACK 로그 저장소 구성

로깅 설명서에서 **LokiStack**은 OpenShift Container Platform 인증 통합을 사용하여 Loki 및 웹 프록시의 로깅 지원 조합을 나타냅니다. LokiStack의 프록시는 OpenShift Container Platform 인증을 사용하여 멀티 테넌시를 적용합니다. **Loki**는 개별 구성 요소 또는 외부 저장소로 로그 저장소를 나타냅니다.

10.3.1. cluster-admin 사용자 역할의 새 그룹 생성



중요

cluster-admin 사용자로 여러 네임스페이스에 대한 애플리케이션 로그를 쿼리합니다. 여기서 클러스터의 모든 네임스페이스 합계는 5120보다 큰 오류입니다: 입력 크기가 너무 긴 (XXXX > 5120) 오류가 발생했습니다. LokiStack의 로그에 대한 액세스를 보다 효과적으로 제어하려면 **cluster-admin** 사용자를 **cluster-admin** 그룹의 멤버로 설정합니다. **cluster-admin** 그룹이 없는 경우 해당 그룹을 생성하고 원하는 사용자를 추가합니다.

다음 절차에 따라 **cluster-admin** 권한이 있는 사용자를 위한 새 그룹을 생성합니다.

절차

1.

다음 명령을 입력하여 새 그룹을 생성합니다.

```
$ oc adm groups new cluster-admin
```

2.

다음 명령을 입력하여 원하는 사용자를 **cluster-admin** 그룹에 추가합니다.

```
$ oc adm groups add-users cluster-admin <username>
```

3.

다음 명령을 입력하여 **cluster-admin** 사용자 역할을 그룹에 추가합니다.

```
$ oc adm policy add-cluster-role-to-group cluster-admin cluster-admin
```

10.3.2. 클러스터를 다시 시작하는 동안 LokiStack 동작

로깅 버전 5.8 이상 버전에서는 **OpenShift Container Platform** 클러스터를 다시 시작할 때 **LokiStack** 수집 및 쿼리 경로는 노드에 사용 가능한 CPU 및 메모리 리소스 내에서 계속 작동합니다. 즉, **OpenShift Container Platform** 클러스터 업데이트 중에 **LokiStack**에 대한 다운 타임이 없습니다. 이 동작은 **PodDisruptionBudget** 리소스를 사용하여 수행됩니다. **Loki Operator**는 특정 조건에서 정상적인 작업을 보장하기 위해 구성 요소별로 사용 가능한 최소 **Pod** 수를 결정하는 **Loki**의 **PodDisruptionBudget** 리소스를 프로비저닝합니다.

추가 리소스



[Pod 중단 예산 Kubernetes 문서](#)

10.3.3. 노드 장애를 허용하도록 Loki 구성

로깅 5.8 이상 버전에서 **Loki Operator**는 **Pod** 유사성 방지 규칙 설정을 지원하여 동일한 구성 요소의 **Pod**가 클러스터의 다른 사용 가능한 노드에 예약되도록 요청합니다.

유사성은 예약할 노드를 제어하는 **Pod**의 속성입니다. 유사성 방지는 **Pod**가 노드에서 예약되지 않도록 하는 **Pod**의 속성입니다.

OpenShift Container Platform에서 **Pod** 유사성 및 **Pod** 유사성 방지를 사용하면 다른 **Pod**의 키 값 라벨에 따라 **Pod**를 예약할 수 있는 노드를 제한할 수 있습니다.

Operator는 **compactor,distributor,gateway,indexGateway,ingester,querier,queryFrontend** 및 **ruler** 구성 요소를 포함하는 모든 **Loki** 구성 요소에 대해 기본 기본 **podAntiAffinity** 규칙을 설정합니다.

requiredDuringSchedulingIgnoredDuringExecution 필드에서 필요한 설정을 구성하여 **Loki** 구성 요소의 기본 **podAntiAffinity** 설정을 덮어쓸 수 있습니다.

ingester 구성 요소에 대한 사용자 설정 예

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
# ...
  template:
    ingester:
      podAntiAffinity:
        # ...
        requiredDuringSchedulingIgnoredDuringExecution: ①
        - labelSelector:
            matchLabels: ②
              app.kubernetes.io/component: ingester
            topologyKey: kubernetes.io/hostname
# ...
```

①

필수 규칙을 정의하는 스텐자입니다.

②

규칙을 적용하려면 일치해야 하는 키-값 쌍(레이블)입니다.

추가 리소스

- [podAntiAffinity v1 코어 Kubernetes 문서](#)
- [노드에 Pod 할당 Kubernetes 설명서](#)
- 유사성 및 유사성 방지 규칙을 사용하여 다른 Pod에 상대적인 Pod 배치

10.3.4. 영역 인식 데이터 복제

로깅 5.8 이상 버전에서 Loki Operator는 Pod 토플로지 분배 제약 조건을 통해 영역 인식 데이터 복제를 지원합니다. 이 기능을 사용하면 단일 영역 장애가 발생할 경우 로그 손실에 대한 안정성과 보호 장치가 향상됩니다. 배포 크기를 **1x.extra.tekton**, **1x.tekton** 또는 **1x.medium**으로 구성하면 **replication.factor** 필드가 자동으로 **2**로 설정됩니다.

적절한 복제를 위해서는 복제 요인에서 지정한 만큼 이상의 가용성 영역이 있어야 합니다. 복제 요인보다 가용성 영역을 더 많이 보유할 수는 있지만 영역이 줄어들면 오류를 작성할 수 있습니다. 각 영역은 최적의 작업을 위해 동일한 수의 인스턴스를 호스팅해야 합니다.

영역 복제가 활성화된 LokiStack CR의 예

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
  replicationFactor: 2 ①
  replication:
    factor: 2 ②
    zones:
      - maxSkew: 1 ③
    topologyKey: topology.kubernetes.io/zone ④
```

①

더 이상 사용되지 않는 필드로 입력된 값은 **replication.factor**로 덮어씁니다.

2

이 값은 배포 크기가 설정 시 선택되면 자동으로 설정됩니다.

3

두 토플로지 도메인 간 최대 Pod 수 차이입니다. 기본값은 1이며 값 0을 지정할 수 없습니다.

4

노드 레이블에 해당하는 토플로지 키 형태로 영역을 정의합니다.

10.3.4.1. 실패한 영역에서 Loki Pod 복구

OpenShift Container Platform에서 특정 가용성 영역 리소스에 액세스할 수 없게 되면 영역 오류가 발생합니다. 가용성 영역은 클라우드 공급자의 데이터 센터 내의 격리된 영역이며 중복성과 내결함성을 강화하기 위한 것입니다. **OpenShift Container Platform** 클러스터가 이를 처리하도록 구성되지 않은 경우 영역 장애가 발생하면 서비스 또는 데이터 손실이 발생할 수 있습니다.

Loki 포드는 **StatefulSet**의 일부이며 **StorageClass** 오브젝트에서 프로비저닝한 PVC(영구 볼륨 클레임)와 함께 제공됩니다. 각 Loki Pod 및 해당 PVC는 동일한 영역에 있습니다. 클러스터에서 영역 오류가 발생하면 **StatefulSet** 컨트롤러에서 실패한 영역에서 영향을 받는 Pod를 자동으로 복구하려고 합니다.



주의

다음 절차에서는 실패한 영역의 PVC와 그 안에 포함된 모든 데이터를 삭제합니다. 완전한 데이터 손실을 방지하려면 LokiStack CR의 복제 요소 필드를 항상 1보다 큰 값으로 설정하여 Loki가 복제되도록 해야 합니다.

사전 요구 사항

-

로깅 버전 5.8 이상

- LokiStack CR에 1보다 큰 복제 인수가 있는지 확인합니다.
- 컨트롤 플레인에서 감지한 영역 장애와 실패한 영역의 노드는 클라우드 공급자 통합으로 표시됩니다.

StatefulSet 컨트롤러는 실패한 영역에서 Pod 일정 변경을 자동으로 시도합니다. 연결된 PVC도 실패한 영역에 있기 때문에 다른 영역으로 자동 일정 조정이 작동하지 않습니다. 새 영역에서 상태 저장 Loki Pod와 프로비저닝된 PVC를 다시 생성할 수 있도록 실패한 영역에서 PVC를 수동으로 삭제해야 합니다.

절차

1. 다음 명령을 실행하여 Pending 상태의 Pod를 나열합니다.

```
oc get pods --field-selector status.phase==Pending -n openshift-logging
```

oc get pods 출력 예

NAME	READY	STATUS	RESTARTS	AGE
logging-loki-index-gateway-1	0/1	Pending	0	17m
logging-loki-ingester-1	0/1	Pending	0	16m
logging-loki-ruler-1	0/1	Pending	0	16m

1

해당 PVC가 실패한 영역에 있기 때문에 이러한 Pod는 Pending 상태입니다.

2. 다음 명령을 실행하여 Pending 상태의 PVC를 나열합니다.

```
oc get pvc -o=json -n openshift-logging | jq '.items[] | select(.status.phase == "Pending") | .metadata.name' -r
```

oc get pvc 출력 예

```
storage-logging-loki-index-gateway-1
storage-logging-loki-ingester-1
wal-logging-loki-ingester-1
storage-logging-loki-ruler-1
wal-logging-loki-ruler-1
```

3.

다음 명령을 실행하여 Pod의 PVC를 삭제합니다.

```
oc delete pvc __<pvc_name>__ -n openshift-logging
```

4.

그런 다음 다음 명령을 실행하여 Pod를 삭제합니다.

```
oc delete pod __<pod_name>__ -n openshift-logging
```

이러한 오브젝트가 성공적으로 삭제되면 사용 가능한 영역에서 자동으로 다시 예약해야 합니다.

10.3.4.1.1. 종료 상태의 PVC 문제 해결

PVC 메타데이터 종료자가 [kubernetes.io/pv-protection](#) 으로 설정된 경우 PVC는 삭제 없이 종료 상태로 중단될 수 있습니다. 종료자를 제거하면 PVC가 성공적으로 삭제될 수 있습니다.

1.

아래 명령을 실행하여 각 PVC의 종료자를 제거한 다음 삭제를 다시 시도합니다.

```
oc patch pvc __<pvc_name>__ -p '{"metadata":{"finalizers":[]}}' -n openshift-logging
```

추가 리소스

- [토플로지 분배 제약 조건 Kubernetes 문서](#)
- [Kubernetes 스토리지 설명서.](#)

- [Pod 토플로지 분배 제약 조건을 사용하여 Pod 배치 제어](#)

10.3.5. Loki 로그에 대한 세분화된 액세스

로깅 5.8 이상에서 **Red Hat OpenShift Logging Operator**는 기본적으로 모든 사용자에게 로그에 대한 액세스 권한을 부여하지 않습니다. 관리자는 **Operator**가 업그레이드되고 이전 구성이 적용되지 않는 한 사용자 액세스를 구성해야 합니다. 구성 및 필요에 따라 다음을 사용하여 로그에 대한 미세 액세스를 구성할 수 있습니다.

- 클러스터 전체 정책
- 네임스페이스 범위 정책
- 사용자 정의 관리자 그룹 생성

관리자는 배포에 적합한 역할 바인딩 및 클러스터 역할 바인딩을 생성해야 합니다. **Red Hat OpenShift Logging Operator**는 다음과 같은 클러스터 역할을 제공합니다.

- **cluster-logging-application-view** 는 애플리케이션 로그를 읽을 수 있는 권한을 부여합니다.
- **cluster-logging-infrastructure-view** 는 인프라 로그를 읽을 수 있는 권한을 부여합니다.
- **cluster-logging-audit-view** 는 감사 로그를 읽을 수 있는 권한을 부여합니다.

이전 버전에서 업그레이드한 경우 추가 클러스터 역할 **logging-application-logs-reader** 및 관련 클러스터 역할 바인딩 **logging-all-authenticated-application-logs-reader**는 이전 버전과의 호환성을 제공하여 네임스페이스에서 인증된 모든 사용자 읽기 액세스를 허용합니다.



참고

네임스페이스별 액세스 권한이 있는 사용자는 애플리케이션 로그를 쿼리할 때 네임스페이스를 제공해야 합니다.

10.3.5.1. 클러스터 전체 액세스

클러스터 역할 바인딩 리소스는 클러스터 역할을 참조하고 클러스터 전체 권한을 설정합니다.

클러스터 역할 바인딩 예

```
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: logging-all-application-logs-reader
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-logging-application-view ①
subjects: ②
- kind: Group
  name: system:authenticated
  apiGroup: rbac.authorization.k8s.io
```

1

추가 ClusterRoles 는 cluster-logging-infrastructure-view 및 cluster-logging-audit-view 입니다.

2

이 개체가 적용되는 사용자 또는 그룹을 지정합니다.

10.3.5.2. 네임스페이스가 지정된 액세스

RoleBinding 리소스는 **ClusterRole** 오브젝트와 함께 사용하여 사용자 또는 그룹이 로그에 액세스할 수 있는 네임스페이스를 정의할 수 있습니다.

RoleBinding의 예

```

kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: allow-read-logs
  namespace: log-test-0 ①
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-logging-application-view
subjects:
- kind: User
  apiGroup: rbac.authorization.k8s.io
  name: testuser-0

```

①

이 RoleBinding 이 적용되는 네임스페이스를 지정합니다.

10.3.5.3. 사용자 정의 관리자 그룹 액세스

광범위한 권한이 필요한 사용자가 많은 대규모 배포가 있는 경우 **adminGroup** 필드를 사용하여 사용자 지정 그룹을 생성할 수 있습니다. LokiStack CR의 **adminGroups** 필드에 지정된 그룹의 멤버인 사용자는 관리자로 간주됩니다. admin 사용자는 **cluster-logging-application-view** 역할도 할당한 경우 모든 네임스페이스의 모든 애플리케이션 로그에 액세스할 수 있습니다.

LokiStack CR의 예

```

apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
  tenants:
    mode: openshift-logging ①
    openshift:

```

adminGroups: ②
 - cluster-admin
 - custom-admin-group ③

1

사용자 지정 관리 그룹은 이 모드에서만 사용할 수 있습니다.

2

이 필드에 빈 목록 [] 값을 입력하면 관리자 그룹이 비활성화됩니다.

3

기본 그룹 덮어쓰기(system:cluster-admins,cluster-admin,dedicated-admin)

추가 리소스

- [RBAC를 사용하여 권한 정의 및 적용](#)

10.3.6. CloudEvent를 사용하여 스트림 기반 보존 활성화

로깅 버전 5.6 이상을 사용하면 로그 스트림에 따라 보존 정책을 구성할 수 있습니다. 이러한 규칙은 테넌트별로 또는 둘 다 전역적으로 설정할 수 있습니다. 둘 다 구성하면 글로벌 규칙이 전에 테넌트 규칙이 적용됩니다.

1.

스트림 기반 보존을 활성화하려면 **LokiStack CR**(사용자 정의 리소스)을 생성합니다.

전역 스트림 기반 보존의 예

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
```

```

limits:
global: ①
  retention: ②
    days: 20
  streams:
    - days: 4
      priority: 1
      selector: '{kubernetes_namespace_name=~"test.+"}' ③
    - days: 1
      priority: 1
      selector: '{log_type="infrastructure"}'
managementState: Managed
replicationFactor: 1
size: 1x.small
storage:
  schemas:
    - effectiveDate: "2020-10-11"
      version: v11
  secret:
    name: logging-loki-s3
    type: aws
storageClassName: standard
tenants:
  mode: openshift-logging

```

①

모든 로그 스트림에 대한 보존 정책을 설정합니다. 참고: 이 필드는 오브젝트 스토리지에서 저장된 로그의 보존 기간에는 영향을 미치지 않습니다.

②

이 블록이 CR에 추가될 때 클러스터에서 보존이 활성화됩니다.

③

로그 스트림을 정의하는 데 사용되는 [LogQL 쿼리를 포함합니다.](#)

테넌트당 스트림 기반 보존 예

```

apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki

```

```

namespace: openshift-logging
spec:
limits:
global:
  retention:
    days: 20
tenants: ①
  application:
    retention:
      days: 1
    streams:
      - days: 4
        selector: '{kubernetes_namespace_name=~"test.+"}' ②
infrastructure:
  retention:
    days: 5
  streams:
    - days: 1
      selector: '{kubernetes_namespace_name=~"openshift-cluster.+"}'
managementState: Managed
replicationFactor: 1
size: 1x.small
storage:
  schemas:
    - effectiveDate: "2020-10-11"
      version: v11
  secret:
    name: logging-loki-s3
    type: aws
storageClassName: standard
tenants:
  mode: openshift-logging

```

1

테넌트별 보존 정책을 설정합니다. 유효한 테넌트 유형은 애플리케이션, 감사 및 인프라입니다.

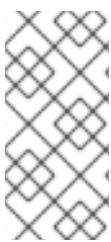
2

로그 스트림을 정의하는 데 사용되는 [LogQL 쿼리를 포함합니다.](#)

2.

LokiStack CR을 적용합니다.

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



참고

이는 저장된 로그의 보존을 관리하기 위한 것이 아닙니다. 지원되는 최대 30일 동안의 저장된 로그의 글로벌 보존 기간은 오브젝트 스토리지로 구성됩니다.

10.3.7. Loki 속도 제한 오류 문제 해결

로그 전달자 API에서 속도 제한을 초과하는 대규모 메시지 블록을 Loki로 전달하면 Loki는 속도 제한 (429) 오류를 생성합니다.

이러한 오류는 정상적인 작동 중에 발생할 수 있습니다. 예를 들어 이미 일부 로그가 있는 클러스터에 로깅을 추가할 때 로깅이 기존 로그 항목을 모두 수집하는 동안 속도 제한 오류가 발생할 수 있습니다. 이 경우 새 로그 추가 속도가 총 속도 제한보다 작으면 기록 데이터가 결국 수집되고 사용자 개입 없이도 속도 제한 오류가 해결됩니다.

속도 제한 오류가 계속 발생하는 경우 **LokiStack CR**(사용자 정의 리소스)을 수정하여 문제를 해결할 수 있습니다.



중요

LokiStack CR은 **Grafana** 호스팅 Loki에서 사용할 수 없습니다. 이는 **Grafana** 호스팅 Loki 서버에는 적용되지 않습니다.

조건

- **Log Forwarder API**는 로그를 Loki로 전달하도록 구성되어 있습니다.
- 시스템에서 2MB보다 큰 메시지 블록을 Loki로 보냅니다. 예를 들면 다음과 같습니다.

```
"values": [{"id": "1630410392689800468", "kind": "Event", "apiVersion": "v1", "version": "1.7.4", "received_at": "2021-08-31T11:46:32.800278+00:00", "@timestamp": "2021-08-31T11:46:32.799692+00:00", "viaq_index_name": "audit-write", "viaq_msg_id": "MzFjYjJKZlItNjY0MC00YWU4LWIwMTEtNGNmM2E5ZmViMGU4", "log_type": "audit"}]}
```

-

`oc logs -n openshift-logging -l component=collector` 를 입력하면 클러스터의 수집기로 그에 다음 오류 메시지 중 하나가 포함된 행이 표시됩니다.

429 Too Many Requests Ingestion rate limit exceeded

Vector 오류 메시지의 예

```
2023-08-25T16:08:49.301780Z WARN sink{component_kind="sink"
component_id=default_loki_infra component_type=loki
component_name=default_loki_infra}: vector::sinks::util::retries: Retrying after error.
error=Server responded with an error: 429 Too Many Requests
internal_log_rate_limit=true
```

Fluentd 오류 메시지의 예

```
2023-08-30 14:52:15 +0000 [warn]: [default_loki_infra] failed to flush the buffer.
retry_times=2 next_retry_time=2023-08-30 14:52:19 +0000
chunk="604251225bf5378ed1567231a1c03b8b"
error_class=Fluent::Plugin::LokiOutput::LogPostError error="429 Too Many Requests
Ingestion rate limit exceeded for user infrastructure (limit: 4194304 bytes/sec) while
attempting to ingest '4082' lines totaling '7820025' bytes, reduce log volume or contact
your Loki administrator to see if the limit can be increased\n"
```

이 오류는 수신 끝점에도 표시됩니다. 예를 들어 **LokiStack** **ingester** Pod에서 다음을 수행합니다.

Loki ingester 오류 메시지의 예

```
level=warn ts=2023-08-30T14:57:34.155592243Z caller=grpc_logging.go:43
duration=1.434942ms method=/logproto.Pusher/Push err="rpc error: code = Code(429)
desc = entry with timestamp 2023-08-30 14:57:32.012778399 +0000 UTC ignored,
reason: 'Per stream rate limit exceeded (limit: 3MB/sec) while attempting to ingest for
stream"
```

절차

- LokiStack CR에서 `ingestionBurstSize` 및 `ingestionRate` 필드를 업데이트합니다.

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
  limits:
    global:
      ingestion:
        ingestionBurstSize: 16 ①
        ingestionRate: 8 ②
# ...
```

1

`ingestionBurstSize` 필드는 배포자 복제본당 최대 로컬 속도 제한 샘플 크기를 **MB**로 정의합니다. 이 값은 하드 제한입니다. 이 값을 단일 푸시 요청에 예상되는 최대 로그 크기로 설정합니다. `ingestionBurstSize` 값보다 큰 단일 요청은 허용되지 않습니다.

2

`ingestionRate` 필드는 초당 수집된 샘플의 최대 양(**MB**)에 대한 소프트 제한입니다. 로그 비율이 제한을 초과하는 경우 속도 제한 오류가 발생하지만 수집기는 로그를 다시 시도합니다. 총 평균이 제한보다 작으면 사용자 개입 없이 시스템을 복구하고 오류가 해결됩니다.

10.3.8. 멤버 목록 생성 실패를 허용하도록 Loki 구성

OpenShift 클러스터에서 관리자는 일반적으로 개인 IP 네트워크 범위를 사용합니다. 결과적으로 LokiStack 멤버 목록 구성은 기본적으로 개인 IP 네트워크만 사용하므로 실패합니다.

관리자는 `memberlist` 구성에 대한 Pod 네트워크를 선택할 수 있습니다. `hashRing` 사양에서 `podIP`를 사용하도록 LokiStack CR을 수정할 수 있습니다. LokiStack CR을 구성하려면 다음 명령을 사용합니다.

```
$ oc patch LokiStack logging-loki -n openshift-logging --type=merge -p '{"spec": {"hashRing": {"memberlist": {"instanceAddrType": "podIP", "type": "memberlist" }}}}'
```

- podIP를 포함하는 LokiStack의 예

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
# ...
  hashRing:
    type: memberlist
    memberlist:
      instanceAddrType: podIP
# ...
```

10.3.9. 추가 리소스

- [Loki 구성 요소 문서](#)
- [Loki 쿼리 언어\(LogQL\) 문서](#)
- [Grafana 대시보드 문서](#)
- [Loki 오브젝트 스토리지 문서](#)
- [Loki Operator IngestionLimitSpec 문서](#)
- [Loki 스토리지 스키마 문서](#)

10.4. ELASTICSEARCH 로그 저장소 구성

Elasticsearch 6을 사용하여 로그 데이터를 저장하고 구성할 수 있습니다.

다음을 포함하여 로그 저장소를 수정할 수 있습니다.

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

10.4.1. 로그 스토리지 구성

ClusterLogging 사용자 정의 리소스(**CR**)를 수정하여 로깅에서 사용하는 로그 스토리지 유형을 구성할 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- **OpenShift CLI(oc)**가 설치되어 있습니다.
- **Red Hat OpenShift Logging Operator**와 **LokiStack** 또는 **Elasticsearch**의 내부 로그 저장소를 설치했습니다.
- **ClusterLogging CR**을 생성했습니다.



참고

로깅 5.9 릴리스에는 업데이트된 OpenShift Elasticsearch Operator 버전이 포함되어 있지 않습니다. 현재 Logging 5.8과 함께 릴리스된 OpenShift Elasticsearch Operator를 사용하는 경우 로깅 5.8의 EOL까지 로깅에서 계속 작동합니다. OpenShift Elasticsearch Operator를 사용하여 기본 로그 스토리지를 관리하는 대신 Loki Operator를 사용할 수 있습니다. 로깅 라이프사이클 날짜에 대한 자세한 내용은 [Platform Agnostic Operators](#) 를 참조하십시오.

절차

1.

ClusterLogging CR logStore 사양을 수정합니다.

ClusterLogging CR 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
# ...
spec:
# ...
logStore:
  type: <log_store_type> ①
  elasticsearch: ②
    nodeCount: <integer>
    resources: {}
    storage: {}
  redundancyPolicy: <redundancy_type> ③
  lokistack: ④
    name: {}
# ...
```

1

로그 저장소 유형을 지정합니다. lokistack 또는 elasticsearch 일 수 있습니다.

2

Elasticsearch 로그 저장소에 대한 선택적 구성 옵션입니다.

3

증복 유형을 지정합니다. 이 값은 **ZeroRedundancy**, **SingleRedundancy**, **MultipleRedundancy** 또는 **FullRedundancy** 일 수 있습니다.

4

LokiStack에 대한 선택적 구성 옵션입니다.

LokiStack을 로그 저장소로 지정하는 **ClusterLogging CR**의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
  name: instance
  namespace: openshift-logging
spec:
  managementState: Managed
  logStore:
    type: lokistack
    lokistack:
      name: logging-loki
# ...
```

2.

다음 명령을 실행하여 **ClusterLogging CR**을 적용합니다.

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

10.4.2. 감사 로그를 로그 저장소로 전달

로깅 배포에서 컨테이너 및 인프라 로그는 기본적으로 **ClusterLogging** 사용자 정의 리소스(**CR**)에 정의된 내부 로그 저장소로 전달됩니다.

감사 로그는 보안 스토리지를 제공하지 않기 때문에 기본적으로 내부 로그 저장소로 전달되지 않습니다. 감사 로그를 전달하는 시스템이 조직 및 정부 규정을 준수하고 올바르게 보호되도록 할 책임이 있습니다.

이 기본 구성이 요구 사항을 충족하는 경우 **ClusterLogForwarder CR**을 구성할 필요가 없습니다. **ClusterLogForwarder CR**이 있는 경우 기본 출력이 포함된 파이프라인을 정의하지 않는 한 로그는 내부 로그 저장소로 전달되지 않습니다.

절차

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

1.

ClusterLogForwarder CR 오브젝트를 정의하는 **YAML** 파일을 생성하거나 편집합니다.

●

모든 로그 유형을 내부 **Elasticsearch** 인스턴스로 보내는 **CR**을 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 CR**이 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

-

[로그 수집 및 전달 정보](#)

10.4.3. 로그 보존 시간 구성

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

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

인덱스가 다음 조건 중 하나와 일치하면 **Elasticsearch**는 현재 인덱스를 이동하고 새 인덱스를 생성하여 인덱스를 를오버합니다.

- 인덱스가 **Elasticsearchh CR**의 **rollover.maxAge** 값보다 오래되었습니다.
- 인덱스 크기가 **40GB × 기본 shard** 수보다 큽니다.
- 인덱스 문서 수가 **40960KB × 기본 shard** 수보다 큽니다.

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

사전 요구 사항

- **Red Hat OpenShift Logging Operator** 및 **OpenShift Elasticsearch Operator**가 설치되어 있어야 합니다.

절차

로그 보존 시간을 구성하려면 다음을 수행합니다.

1.

retentionPolicy 매개변수를 추가하거나 수정하려면 **ClusterLogging CR**을 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 로그는 삭제됩니다. 기본적으로 로그는 7일 동안 유지됩니다.

2.

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

예를 들어 **Red Hat OpenShift Logging Operator**가 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 Platform**은 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

2

OpenShift Container Platform이 룰오버된 인덱스를 삭제하는 경우 이 설정은 **ClusterLogging CR**에서 설정한 **maxAge**입니다.

3

인덱스를 룰오버할 때 고려해야 할 **OpenShift Container Platform**의 인덱스 수명입니다. 이 값은 **ClusterLogging CR**에서 설정한 **maxAge**에서 결정됩니다.

4

OpenShift Container Platform에서 인덱스를 룰오버해야 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

10.4.4. 로그 저장소에 대한 CPU 및 메모리 요청 구성

각 구성 요소 사양을 통해 CPU 및 메모리 요청을 조정할 수 있습니다. **OpenShift Elasticsearch Operator**가 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 **OOMKilled**로 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

- Red Hat OpenShift Logging 및 Elasticsearch Operator가 설치되어 있어야 합니다.

절차

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

Pod에서 사용할 수 있는 최대 리소스 양입니다.

3

Pod를 예약하는 데 필요한 최소 리소스입니다.

4

필요에 따라 Elasticsearch 프록시에 대한 CPU 및 메모리 요청을 지정합니다. 이 값을 비워 두면 OpenShift Elasticsearch Operator가 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 최소 수량입니다.

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.

10.4.5. 로그 저장소에 대한 복제 정책 구성

Elasticsearch shard가 클러스터의 데이터 노드에 복제되는 방법을 정의할 수 있습니다.

사전 요구 사항

- Red Hat OpenShift Logging 및 Elasticsearch Operator가 설치되어 있어야 합니다.

절차

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 **shard**를 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 **MultipleRedundancy**보다 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.

10.4.6. Elasticsearch Pod 축소

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



참고

Elasticsearch 클러스터가 ZeroRedundancy로 설정된 경우 Elasticsearch Pod를 축소해서는 안 됩니다.

10.4.7. 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 **Gluster**와 같은 NAS를 통해)으로 사용할 수 없습니다. 데이터 손상 및 기타 문제가 발생할 수 있습니다.

사전 요구 사항

-

Red Hat OpenShift Logging 및 **Elasticsearch Operator**가 설치되어 있어야 합니다.

절차

1.

ClusterLogging CR을 편집하여 클러스터의 각 데이터 노드가 영구 볼륨 클레임에 바인딩되도록 지정합니다.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
# ...
spec:
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      storage:
        storageClassName: "gp2"
        size: "200G"
```

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



참고

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

10.4.8. 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

- Red Hat OpenShift Logging 및 Elasticsearch Operator가 설치되어 있어야 합니다.

절차

- emptyDir을 지정 하려면 ClusterLogging CR을 편집합니다.

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

10.4.9. 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

- Red Hat OpenShift Logging 및 Elasticsearch Operator가 설치되어 있어야 합니다.

절차

클러스터를 롤링 재시작하려면 다음을 수행합니다.

1. **openshift-logging** 프로젝트로 변경합니다.

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

2. **Elasticsearch pod**의 이름을 가져옵니다.

```
$ oc get pods -l component=elasticsearch
```

3. **Elasticsearch**로 새 로그 전송을 중지하도록 수집기 Pod를 축소합니다.

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

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

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

예를 들면 다음과 같습니다.

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

출력 예

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

5. **OpenShift Container Platform es_util** 도구를 사용하여 의도적으로 노드를 중단할 때

shard 밸런싱을 방지합니다.

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

예를 들면 다음과 같습니다.

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

출력 예

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

6.

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

a.

기본적으로 **OpenShift Container Platform Elasticsearch** 클러스터는 노드에 대한 롤아웃을 차단합니다. 다음 명령을 사용하여 롤아웃을 허용하고 **Pod**가 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
```

세 Pod가 배포되었습니다. Pod에 컨테이너가 준비되면 다음 배포로 이동할 수 있습니다.

```
$ oc get pods -l component=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 룰아웃을 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 Pod**에서 롤아웃을 수행한 경우 그 **Pod**는 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 Pod**에 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.

새 로그를 Elasticsearch로 보내도록 수집기 Pod를 확장합니다.

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

10.4.10. 로그 저장소 서비스를 경로로 노출

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

외부에서는 재암호화 경로, **OpenShift Container Platform** 토큰 및 설치된 로그 저장소 CA 인증서를 생성하여 로그 저장소에 액세스할 수 있습니다. 그런 후 다음을 포함하는 cURL 요청으로 로그 저장소 서

비스를 호스팅하는 노드에 액세스합니다.

- **Authorization: Bearer \${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 Current
          Dload Upload Total Spent Left Speed
100  29 100  29  0  0 108  0 --:--:-- --:--:-- --:--:-- 108
```

사전 요구 사항

- Red Hat OpenShift Logging 및 Elasticsearch Operator가 설치되어 있어야 합니다.
- 로그에 액세스하려면 프로젝트에 액세스할 수 있어야 합니다.

절차

로그 저장소를 외부에 노출하려면 다음을 수행합니다.

1. openshift-logging 프로젝트로 변경합니다.

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

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

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

출력 예

```
admin-ca
```

3. 로그 저장소 서비스의 경로를 YAML 파일로 생성합니다.

a.

다음을 사용하여 YAML 파일을 생성합니다.

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

①

로그 저장소 CA 인증서를 추가하거나 다음 단계에서 명령을 사용합니다. 일부 재암호화 경로에 필요한 `spec.tls.key`, `spec.tls.certificate` 및 `spec.tls.caCertificate` 매개변수를 설정할 필요는 없습니다.

b.

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

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

c.

경로를 생성합니다.

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

출력 예

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

4.

Elasticsearch 서비스가 노출되어 있는지 확인합니다.

a.

요청에 사용할 이 서비스 계정의 토큰을 가져옵니다.

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

b.

생성한 **elasticsearch** 경로를 환경 변수로 설정합니다.

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

c.

경로가 성공적으로 생성되었는지 확인하려면 노출된 경로를 통해 **Elasticsearch**에 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>"
}
```

10.4.11. 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. default를 지정하는 outputRefs 요소가 없는지 확인합니다. 예를 들면 다음과 같습니다.

outputRefs:
- default



주의

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

절차

1.

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

\$ oc edit ClusterLogging instance

2.

ClusterLogging CR에서 logStore, visualization 스템자를 제거하십시오.

3.

ClusterLogging CR의 collection 스템자를 유지합니다. 결과는 다음 예와 유사해야 합니다.

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

```
metadata:  
  name: "instance"  
  namespace: "openshift-logging"  
spec:  
  managementState: "Managed"  
  collection:  
    type: "fluentd"  
    fluentd: {}
```

4. 수집기 Pod가 재배포되었는지 확인합니다.

```
$ oc get pods -l component=collector -n openshift-logging
```

11장. 로깅 경고

11.1. 기본 로깅 경고

로깅 경고는 **Red Hat OpenShift Logging Operator** 설치의 일부로 설치됩니다. 경고는 로그 수집 및 로그 스토리지 백엔드에서 내보낸 메트릭에 따라 달라집니다. **Red Hat OpenShift Logging Operator**를 설치할 때 이 네임스페이스에서 **Operator** 권장 클러스터 모니터링을 활성화하는 옵션을 선택한 경우 이러한 메트릭이 활성화 됩니다. 로깅 **Operator** 설치에 대한 자세한 내용은 [웹 콘솔을 사용하여 로깅 설치를 참조하십시오.](#)

로컬 **Alertmanager** 인스턴스를 비활성화하지 않은 경우 기본 로깅 경고는 **openshift-monitoring** 네임스페이스의 **OpenShift Container Platform** 모니터링 스택 **Alertmanager**로 전송됩니다.

11.1.1. 관리자 및 개발자 관점에서 경고 UI에 액세스

경고 UI는 **OpenShift Container Platform** 웹 콘솔의 관리자 관점 및 개발자 화면을 통해 액세스할 수 있습니다.

- 관리자 관점에서 모니터링 → 경고로 이동합니다. 이 관점에서 경고 UI의 세 가지 주요 페이지는 경고, 음소거 및 경고 규칙 페이지입니다.
- 개발자 관점에서 **Observe** → <project_name> → 경고로 이동합니다. 이 관점에서 경고, 음소거 및 경고 규칙은 모두 경고 페이지에서 관리됩니다. 경고 페이지에 표시된 결과는 선택한 프로젝트에 특정적입니다.



참고

개발자 관점에서는 프로젝트: <project_name> 목록에서 액세스할 수 있는 코어 **OpenShift Container Platform** 및 사용자 정의 프로젝트에서 선택할 수 있습니다. 그러나 **cluster-admin** 권한이 없는 경우 핵심 **OpenShift Container Platform** 프로젝트와 관련된 경고, 음소거, 경고 규칙이 표시되지 않습니다.

11.1.2. 벡터 수집기 경고

로깅 5.7 이상 버전에서는 벡터 수집기에서 다음 경고가 생성됩니다. **OpenShift Container Platform** 웹 콘솔에서 이러한 경고를 볼 수 있습니다.

표 11.1. 벡터 수집기 경고

경고	메시지	설명	심각도
CollectorHighErrorRate	레코드의 <value>는 vector <instance>에 의해 오류가 발생했습니다.	벡터 출력 오류 수는 기본적으로 이전 15분 동안 10개 이상입니다.	경고
CollectorNodeDown	Prometheus 는 10m 이상 벡터 <instance>를 스크랩 할 수 없습니다.	벡터는 Prometheus가 특정 Vector 인스턴스를 스크랩 할 수 없다고 보고합니다.	심각
CollectorVeryHighErrorRate	레코드의 <value>는 vector <instance>에 의해 오류가 발생했습니다.	벡터 구성 요소 오류의 수는 기본적으로 이전 15분 동안 25 개를 초과합니다.	심각
FluentdQueueLengthIncreasing	마지막 1h에서 fluentd <instance> 버퍼 큐 길이는 1보다 지속적으로 증가 했습니다. 현재 값은 <value>입니다.	Fluentd는 큐 크기가 증가하고 있다고 보고합니다.	경고

11.1.3. Fluentd 수집기 경고

다음 경고는 레거시 **Fluentd** 로그 수집기에 의해 생성됩니다. **OpenShift Container Platform** 웹 콘솔에서 이러한 경고를 볼 수 있습니다.

표 11.2. Fluentd 수집기 경고

경고	메시지	설명	심각도
FluentDHighErrorRate	fluentd <instance>에 의해 레코드의 <value>에서 오류가 발생했습니다.	FluentD 출력 오류의 수는 높 으며 기본적으로 이전 15분 동안 10개 이상입니다.	경고
FluentdNodeDown	Prometheus 는 fluentd <instance>를 10분 이상 스크랩 할 수 없습니다.	Fluentd는 Prometheus가 특정 Fluentd 인스턴스를 스크 랙 할 수 없다고 보고했습니다.	심각
FluentdQueueLengthIncreasing	마지막 1h에서 fluentd <instance> 버퍼 큐 길이는 1보다 지속적으로 증가 했습니다. 현재 값은 <value>입니다.	Fluentd는 큐 크기가 증가하고 있다고 보고합니다.	경고

경고	메시지	설명	심각도
FluentDVeryHighErrorRate	fluentd <instance>에 의해 레코드의 <value>에서 오류가 발생했습니다.	FluentD 출력 오류의 수는 기본적으로 이전 15분 동안 25개 이상으로 매우 높습니다.	심각

11.1.4. Elasticsearch 경고 규칙

OpenShift Container Platform 웹 콘솔에서 이러한 경고 규칙을 볼 수 있습니다.

표 11.3. 경고 규칙

경고	설명	심각도
ElasticsearchClusterNotHealthy	클러스터 상태가 2분 이상 빨간색이었습니다. 클러스터가 쓰기를 허용하지 않거나 shard가 누락되었거나 마스터 노드가 아직 선택되지 않았을 수 있습니다.	심각
ElasticsearchClusterNotHealthy	클러스터 상태가 최소 20분 동안 노란색이었습니다. 일부 shard 복제본이 할당되지 않았습니다.	경고
ElasticsearchDiskSpaceRunningLow	클러스터는 향후 6시간 내에 디스크 공간이 부족할 것으로 예상됩니다.	심각
ElasticsearchHighFileDescriptorUsage	클러스터는 다음 시간 내에 파일 설명자가 없을 것으로 예상됩니다.	경고
ElasticsearchJVMHeapUsageHigh	지정된 노드의 JVM 힙 사용량이 높습니다.	경고
ElasticsearchNodeDiskWatermarkReached	디스크 여유 공간이 부족하여 지정된 노드가 낮은 워터마크에 도달했습니다. 더 이상 shard를 이 노드에 할당할 수 없습니다. 노드에 디스크 공간을 추가하는 것을 고려해야 합니다.	정보
ElasticsearchNodeDiskWatermarkReached	디스크 여유 공간이 부족하여 지정된 노드가 높은 워터마크에 도달했습니다. 일부 shard는 가능한 경우 다른 노드에 다시 할당됩니다. 노드에 디스크 공간을 더 추가하거나 이 노드에 할당된 오래된 인덱스를 삭제하십시오.	경고
ElasticsearchNodeDiskWatermarkReached	디스크 여유 공간이 부족하여 지정된 노드가 플러드 워터마크에 도달했습니다. 이 노드에 할당된 shard가 있는 모든 인덱스에는 읽기 전용 블록이 적용됩니다. 디스크 사용량이 높은 워터마크 아래로 떨어지면 인덱스 블록을 수동으로 해제해야 합니다.	심각
ElasticsearchJVMHeapUsageHigh	지정된 노드의 JVM 힙 사용량이 너무 높습니다.	경고

경고	설명	심각도
ElasticsearchWriteRequestsRejectionJumps	Elasticsearch의 지정된 노드에서 쓰기 거부가 증가하고 있습니다. 이 노드는 인덱싱 속도를 따라가지 못할 수 있습니다.	경고
AggregatedLoggingSystemCPUHigh	지정된 노드의 시스템에서 사용하는 CPU가 너무 높습니다.	경고
ElasticsearchProcessCPUHigh	지정된 노드에서 Elasticsearch가 사용하는 CPU가 너무 높습니다.	경고

11.1.5. 추가 리소스



[코어 플랫폼 경고 규칙 설정](#)

12장. 성능 및 안정성 튜닝

12.1. 흐름 제어 메커니즘

로그를 수집할 수 있는 것보다 빠르게 생성되는 경우 출력으로 전송되는 로그 볼륨을 예측하거나 제어하기 어려울 수 있습니다. 출력으로 전송되는 로그 볼륨을 예측하거나 제어할 수 없으므로 로그가 손실될 수 있습니다. 시스템 중단 및 로그 버퍼가 사용자 제어 없이 누적되면 연결이 복원될 때 긴 복구 시간과 높은 대기 시간이 발생할 수 있습니다.

관리자는 로깅에 대한 흐름 제어 메커니즘을 구성하여 로깅 속도를 제한할 수 있습니다.

12.1.1. 흐름 제어 메커니즘의 이점

- 비용 및 볼륨 로깅은 보다 정확하게 예측할 수 있습니다.
- **Noisy** 컨테이너는 다른 컨테이너를 중단하는 바인딩되지 않은 로그 트래픽을 생성할 수 없습니다.
- 낮은 값 로그를 무시하면 로깅 인프라의 로드가 줄어듭니다.
- 높은 값 로그는 높은 속도 제한을 할당하여 낮은 값 로그보다 선호될 수 있습니다.

12.1.2. 속도 제한 구성

속도 제한은 수집기별로 구성됩니다. 즉, 최대 로그 수집 속도는 속도 제한을 곱한 수집기 인스턴스 수입니다.

로그는 각 노드의 파일 시스템에서 수집되므로 수집기는 각 클러스터 노드에 배포됩니다. 예를 들어, 수집기당 초당 최대 속도 제한이 있는 3-노드 클러스터에서는 최대 로그 수집 속도는 초당 **30개**입니다.

출력에 기록된 레코드의 정확한 바이트 크기는 변환, 다른 인코딩 또는 기타 요인으로 인해 달라질 수 있으므로 속도 제한은 바이트 대신 레코드 수로 설정됩니다.

다음 두 가지 방법으로 **ClusterLogForwarder CR**(사용자 정의 리소스)에서 속도 제한을 구성할 수 있습니다.

출력 속도 제한

출력의 네트워크 또는 스토리지 용량과 일치하도록 아웃바운드 로그의 속도를 선택한 출력으로 제한합니다. 출력 속도 제한은 집계된 출력당 속도를 제어합니다.

입력 속도 제한

선택한 컨테이너에 대한 로그 컬렉션의 컨테이너당 속도를 제한합니다.

12.1.3. 로그 전달자 출력 속도 제한 구성

ClusterLogForwarder 사용자 정의 리소스(CR)를 구성하여 아웃바운드 로그 속도를 지정된 출력으로 제한할 수 있습니다.

사전 요구 사항

- **Red Hat OpenShift Logging Operator**가 설치되어 있습니다.
- 관리자 권한이 있습니다.

절차

1. 지정된 출력의 **ClusterLogForwarder CR**에 **maxRecordsPerSecond** 제한 값을 추가합니다.

다음 예제에서는 **kafka-example**이라는 Kafka 브로커 출력에 대해 수집기 출력 속도 제한을 구성하는 방법을 보여줍니다.

ClusterLogForwarder CR의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
# ...
spec:
```

```
# ...
outputs:
- name: kafka-example ①
  type: kafka ②
  limit:
    maxRecordsPerSecond: 1000000 ③
# ...
```

1

출력 이름입니다.

2

출력 유형입니다.

3

로그 출력 속도 제한입니다. 이 값은 초당 Kafka 브로커로 보낼 수 있는 최대 로그의 최대값을 설정합니다. 이 값은 기본적으로 설정되지 않습니다. 기본 동작은 죄선의 것이며 로그 전달자가 유지할 수 없는 경우 레코드가 삭제됩니다. 이 값이 0 이면 로그가 전달되지 않습니다.

2.

ClusterLogForwarder CR을 적용합니다.

명령 예

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

추가 리소스

•

로그 출력 유형

12.1.4. 로그 전달자 입력 속도 제한 구성

ClusterLogForwarder CR(사용자 정의 리소스)을 구성하여 수집되는 들어오는 로그의 속도를 제한할 수 있습니다. 컨테이너당 또는 네임스페이스별로 입력 제한을 설정할 수 있습니다.

사전 요구 사항

- Red Hat OpenShift Logging Operator가 설치되어 있습니다.
- 관리자 권한이 있습니다.

절차

1.

지정된 입력의 경우 **maxRecordsPerSecond** 제한 값을 **ClusterLogForwarder CR**에 추가 합니다.

다음 예제에서는 다양한 시나리오에 대한 입력 속도 제한을 구성하는 방법을 보여줍니다.

특정 라벨을 사용하여 컨테이너에 대한 컨테이너당 제한을 설정하는 **ClusterLogForwarder CR**의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
# ...
spec:
# ...
inputs:
- name: <input_name> ①
application:
selector:
matchLabels: { example: label } ②
containerLimit:
maxRecordsPerSecond: 0 ③
# ...
```

1

입력 이름입니다.

2

레이블 목록입니다. 이러한 라벨이 Pod에 적용되는 라벨과 일치하면 maxRecordsPerSecond 필드에 지정된 컨테이너별 제한이 해당 컨테이너에 적용됩니다.

3

속도 제한을 구성합니다. maxRecordsPerSecond 필드를 0으로 설정하면 컨테이너에 대한 로그가 수집되지 않습니다. maxRecordsPerSecond 필드를 다른 값으로 설정하면 초당 최대 레코드 수가 컨테이너에 수집됩니다.

선택한 네임스페이스에서 컨테이너에 대한 컨테이너당 제한을 설정하는 ClusterLogForwarder CR의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
# ...
spec:
# ...
inputs:
- name: <input_name> ①
  application:
    namespaces: [ example-ns-1, example-ns-2 ] ②
    containerLimit:
      maxRecordsPerSecond: 10 ③
- name: <input_name>
  application:
    namespaces: [ test ]
    containerLimit:
      maxRecordsPerSecond: 1000
# ...
```

1

입력 이름입니다.

2

3

속도 제한을 구성합니다. **maxRecordsPerSecond** 필드를 10 으로 설정하면 나열된 네임스페이스의 각 컨테이너에 대해 초당 최대 10개의 레코드가 수집됩니다.

2.

ClusterLogForwarder CR을 적용합니다.

명령 예

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

13장. 리소스 예약

13.1. 노드 선택기를 사용하여 로깅 리소스 이동

노드 선택기는 노드의 사용자 정의 라벨 및 Pod에 지정된 선택기를 사용하여 정의한 키/값 쌍으로 구성된 맵을 지정합니다.

노드에서 Pod를 실행하려면 노드의 라벨과 동일한 키/값 노드 선택기가 Pod에 있어야 합니다.

13.1.1. 노드 선택기 정보

Pod의 노드 선택기와 노드의 라벨을 사용하여 Pod가 예약되는 위치를 제어할 수 있습니다. 노드 선택기를 사용하면 OpenShift Container Platform에서 일치하는 라벨이 포함된 노드에 Pod를 예약합니다.

노드 선택기를 사용하여 특정 노드에 특정 Pod를 배치하고, 클러스터 수준 노드 선택기를 사용하여 클러스터의 특정 노드에 새 Pod를 배치하고, 프로젝트 노드 선택기를 사용하여 특정 노드의 프로젝트에 새 Pod를 배치할 수 있습니다.

예를 들어 클러스터 관리자는 애플리케이션 개발자가 생성하는 모든 Pod에 노드 선택기를 포함하여 지리적으로 가장 가까운 노드에만 Pod를 배포할 수 있는 인프라를 생성할 수 있습니다. 이 예제에서 클러스터는 두 지역에 분배된 데이터센터 5개로 구성됩니다. 미국에서는 노드의 라벨을 us-east, us-central 또는 us-west로 지정합니다. 아시아 태평양 지역(APAC)에서는 노드의 라벨을 apac-east 또는 apac-west로 지정합니다. 개발자는 생성한 Pod에 노드 선택기를 추가하여 해당 노드에 Pod가 예약되도록 할 수 있습니다.

Pod 오브젝트에 노드 선택기가 포함되어 있지만 일치하는 라벨이 있는 노드가 없는 경우 Pod를 예약하지 않습니다.



중요

동일한 Pod 구성의 노드 선택기 및 노드 유사성을 사용 중인 경우 다음 규칙에서 노드에 대한 Pod 배치를 제어합니다.

- nodeSelector와 nodeAffinity를 둘 다 구성하는 경우 Pod를 후보 노드에 예약하기 위해서는 두 상태를 모두 충족해야 합니다.
- nodeAffinity 유형과 연결된 nodeSelectorTerms를 여러 개 지정하는 경우 nodeSelectorTerms 중 하나를 충족하면 Pod를 노드에 예약할 수 있습니다.
- nodeSelectorTerms와 연결된 matchExpressions를 여러 개 지정하는 경우 모든 matchExpressions를 충족할 때만 Pod를 노드에 예약할 수 있습니다.

특정 Pod 및 노드의 노드 선택기

노드 선택기 및 라벨을 사용하여 특정 Pod가 예약된 노드를 제어할 수 있습니다.

노드 선택기와 라벨을 사용하려면 먼저 Pod의 일정이 조정되지 않도록 노드에 라벨을 지정한 다음 노드 선택기를 Pod에 추가합니다.



참고

예약된 기준 Pod에 노드 선택기를 직접 추가할 수 없습니다. 배포 구성과 같이 Pod를 제어하는 오브젝트에 라벨을 지정해야 합니다.

예를 들어 다음 Node 오브젝트에는 region: east 라벨이 있습니다.

라벨이 있는 Node 오브젝트 샘플

```
kind: Node
apiVersion: v1
metadata:
  name: ip-10-0-131-14.ec2.internal
  selfLink: /api/v1/nodes/ip-10-0-131-14.ec2.internal
```

```

uid: 7bc2580a-8b8e-11e9-8e01-021ab4174c74
resourceVersion: '478704'
creationTimestamp: '2019-06-10T14:46:08Z'
labels:
  kubernetes.io/os: linux
  failure-domain.beta.kubernetes.io/zone: us-east-1a
  node.openshift.io/os_version: '4.5'
  node-role.kubernetes.io/worker: ""
  failure-domain.beta.kubernetes.io/region: us-east-1
  node.openshift.io/os_id: rhcos
  beta.kubernetes.io/instance-type: m4.large
  kubernetes.io/hostname: ip-10-0-131-14
  beta.kubernetes.io/arch: amd64
  region: east ①
  type: user-node
#...

```

①

Pod 노드 선택기와 일치해야 하는 라벨입니다.

Pod에는 **type: user-node,region: east** 노드 선택기가 있습니다.

노드 선택기가 있는 **Pod** 오브젝트 샘플

```

apiVersion: v1
kind: Pod
metadata:
  name: s1
#...
spec:
  nodeSelector: ①
  region: east
  type: user-node
#...

```

①

노드 라벨과 일치해야 하는 노드 선택기입니다. 노드에는 각 노드 선택기에 대한 레이블이 있어야 합니다.

예제 Pod 사양을 사용하여 Pod를 생성하면 예제 노드에 예약할 수 있습니다.

기본 클러스터 수준 노드 선택기

기본 클러스터 수준 노드 선택기를 사용하면 해당 클러스터에서 Pod를 생성할 때 OpenShift Container Platform에서 기본 노드 선택기를 Pod에 추가하고 일치하는 라벨을 사용하여 노드에서 Pod를 예약합니다.

예를 들어 다음 Scheduler 오브젝트에는 기본 클러스터 수준 `region=east` 및 `type=user-node` 노드 선택기가 있습니다.

스케줄러 Operator 사용자 정의 리소스의 예

```
apiVersion: config.openshift.io/v1
kind: Scheduler
metadata:
  name: cluster
  #...
spec:
  defaultNodeSelector: type=user-node,region=east
  #...
```

해당 클러스터의 노드에는 `type=user-node,region=east` 라벨이 있습니다.

Node 오브젝트의 예

```
apiVersion: v1
kind: Node
metadata:
  name: ci-1n-qg1il3k-f76d1-hlmhl-worker-b-df2s4
  #...
  labels:
    region: east
    type: user-node
  #...
```

노드 선택기가 있는 Pod 오브젝트의 예

```
apiVersion: v1
kind: Pod
metadata:
  name: s1
#...
spec:
  nodeSelector:
    region: east
#...
```

예제 클러스터에서 예제 Pod 사양을 사용하여 Pod를 생성하면 Pod가 클러스터 수준 노드 선택기와 함께 생성되어 라벨이 지정된 노드에 예약됩니다.

Pod가 라벨이 지정된 노드에 있는 Pod 목록의 예

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
NOMINATED NODE		READINESS	GATES			
pod-s1	1/1	Running	0	20s	10.131.2.6	ci-1n-qg1il3k-f76d1-hlmhl-worker-b-df2s4
<none>		<none>				

참고

Pod를 생성하는 프로젝트에 프로젝트 노드 선택기가 있는 경우 해당 선택기가 클러스터 수준 노드 선택기보다 우선합니다. Pod에 프로젝트 노드 선택기가 없으면 Pod가 생성되거나 예약되지 않습니다.

프로젝트 노드 선택기를 사용하면 이 프로젝트에서 Pod를 생성할 때 OpenShift Container Platform에서 Pod에 노드 선택기를 추가하고 라벨이 일치하는 노드에 Pod를 예약합니다. 클러스터 수준 기본 노드 선택기가 있는 경우 프로젝트 노드 선택기가 우선합니다.

예를 들어 다음 프로젝트에는 `region=east` 노드 선택기가 있습니다.

Namespace 오브젝트의 예

```
apiVersion: v1
kind: Namespace
metadata:
  name: east-region
  annotations:
    openshift.io/node-selector: "region=east"
#...
```

다음 노드에는 `type=user-node,region=east` 라벨이 있습니다.

Node 오브젝트의 예

```
apiVersion: v1
kind: Node
metadata:
  name: ci-in-qg1il3k-f76d1-hlmhl-worker-b-df2s4
#...
  labels:
    region: east
    type: user-node
#...
```

이 예제 프로젝트에서 예제 Pod 사양을 사용하여 Pod를 생성하면 Pod가 프로젝트 노드 선택기와 함께 생성되어 라벨이 지정된 노드에 예약됩니다.

Pod 오브젝트의 예

```

apiVersion: v1
kind: Pod
metadata:
  namespace: east-region
#...
spec:
  nodeSelector:
    region: east
    type: user-node
#...

```

Pod가 라벨이 지정된 노드에 있는 Pod 목록의 예

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
NOMINATED NODE		READINESS	GATES			
pod-s1	1/1	Running	0	20s	10.131.2.6	ci-1n-qg1il3k-f76d1-hlmhl-worker-b-df2s4
<none>		<none>				

Pod에 다른 노드 선택기가 포함된 경우 프로젝트의 Pod가 생성되거나 예약되지 않습니다. 예를 들어 다음 Pod를 예제 프로젝트에 배포하면 Pod가 생성되지 않습니다.

노드 선택기가 유효하지 않은 Pod 오브젝트의 예

```

apiVersion: v1
kind: Pod
metadata:
  name: west-region
#...
spec:
  nodeSelector:
    region: west
#...

```

13.1.2. Loki Pod 배치

Pod의 허용 오차 또는 노드 선택기를 사용하여 **Loki Pod**가 실행되는 노드를 제어하고 다른 워크로드가 해당 노드를 사용하지 못하도록 할 수 있습니다.

LokiStack CR(사용자 정의 리소스)을 사용하여 로그 저장소 **Pod**에 허용 오차를 적용하고 노드 사양이 있는 노드에 **taint**를 적용할 수 있습니다. 노드의 테인트는 테인트를 허용하지 않는 모든 **Pod**를 거절하도록 노드에 지시하는 키:값 쌍입니다. 다른 **Pod**에 없는 특정 키:값 쌍을 사용하면 해당 노드에서 로그 저장소 **Pod**만 실행할 수 있습니다.

노드 선택기가 있는 **LokiStack**의 예

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
# ...
  template:
    compactor: ①
    nodeSelector:
      node-role.kubernetes.io/infra: "" ②
    distributor:
      nodeSelector:
        node-role.kubernetes.io/infra: """
    gateway:
      nodeSelector:
        node-role.kubernetes.io/infra: """
    indexGateway:
      nodeSelector:
        node-role.kubernetes.io/infra: """
    ingester:
      nodeSelector:
        node-role.kubernetes.io/infra: """
    querier:
      nodeSelector:
        node-role.kubernetes.io/infra: """
    queryFrontend:
      nodeSelector:
        node-role.kubernetes.io/infra: """
    ruler:
```

```

nodeSelector:
  node-role.kubernetes.io/infra: ""
# ...

```

1

노드 선택기에 적용되는 구성 요소 Pod 유형을 지정합니다.

2

정의된 라벨이 포함된 노드로 이동되는 Pod를 지정합니다.

이전 예제 구성에서 모든 Loki Pod는 node-role.kubernetes.io/infra: "" 라벨이 포함된 노드로 이동합니다.

노드 선택기 및 허용 오차가 있는 LokiStack CR의 예

```

apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
# ...
template:
  compactor:
    nodeSelector:
      node-role.kubernetes.io/infra: ""
    tolerations:
      - effect: NoSchedule
        key: node-role.kubernetes.io/infra
        value: reserved
      - effect: NoExecute
        key: node-role.kubernetes.io/infra
        value: reserved
  distributor:
    nodeSelector:
      node-role.kubernetes.io/infra: ""
    tolerations:
      - effect: NoSchedule
        key: node-role.kubernetes.io/infra
        value: reserved
      - effect: NoExecute

```

```
key: node-role.kubernetes.io/infra
value: reserved
nodeSelector:
  node-role.kubernetes.io/infra: ""
tolerations:
- effect: NoSchedule
  key: node-role.kubernetes.io/infra
  value: reserved
- effect: NoExecute
  key: node-role.kubernetes.io/infra
  value: reserved
indexGateway:
nodeSelector:
  node-role.kubernetes.io/infra: ""
tolerations:
- effect: NoSchedule
  key: node-role.kubernetes.io/infra
  value: reserved
- effect: NoExecute
  key: node-role.kubernetes.io/infra
  value: reserved
ingester:
nodeSelector:
  node-role.kubernetes.io/infra: ""
tolerations:
- effect: NoSchedule
  key: node-role.kubernetes.io/infra
  value: reserved
- effect: NoExecute
  key: node-role.kubernetes.io/infra
  value: reserved
querier:
nodeSelector:
  node-role.kubernetes.io/infra: ""
tolerations:
- effect: NoSchedule
  key: node-role.kubernetes.io/infra
  value: reserved
- effect: NoExecute
  key: node-role.kubernetes.io/infra
  value: reserved
queryFrontend:
nodeSelector:
  node-role.kubernetes.io/infra: ""
tolerations:
- effect: NoSchedule
  key: node-role.kubernetes.io/infra
  value: reserved
- effect: NoExecute
  key: node-role.kubernetes.io/infra
  value: reserved
ruler:
nodeSelector:
  node-role.kubernetes.io/infra: ""
tolerations:
- effect: NoSchedule
```

```

key: node-role.kubernetes.io/infra
value: reserved
- effect: NoExecute
  key: node-role.kubernetes.io/infra
  value: reserved
gateway:
nodeSelector:
  node-role.kubernetes.io/infra: ""
tolerations:
- effect: NoSchedule
  key: node-role.kubernetes.io/infra
  value: reserved
- effect: NoExecute
  key: node-role.kubernetes.io/infra
  value: reserved
# ...

```

LokiStack(CR)의 `nodeSelector` 및 허용 오차 필드를 구성하려면 `oc explain` 명령을 사용하여 특정 리소스에 대한 설명 및 필드를 볼 수 있습니다.

```
$ oc explain lokistack.spec.template
```

출력 예

```

KIND: LokiStack
VERSION: loki.grafana.com/v1

RESOURCE: template <Object>

DESCRIPTION:
  Template defines the resource/limits/tolerations/nodeselectors per
  component

FIELDS:
  compactor <Object>
    Compactor defines the compaction component spec.

  distributor <Object>
    Distributor defines the distributor component spec.

...

```

자세한 내용은 특정 필드를 추가할 수 있습니다.

```
$ oc explain lokistack.spec.template.compactor
```

출력 예

KIND: LokiStack
VERSION: loki.grafana.com/v1

RESOURCE: compactor <Object>

DESCRIPTION:

Compactor defines the compaction component spec.

FIELDS:

nodeSelector <map[string]string>

NodeSelector defines the labels required by a node to schedule the component onto it.

...

13.1.3. 로깅 수집기에 대한 리소스 구성 및 스케줄링

관리자는 동일한 네임스페이스에 있고 지원되는 ClusterLogForwarder CR과 동일한 이름이 있는 ClusterLogging 사용자 정의 리소스(CR)를 생성하여 수집기의 리소스 또는 스케줄링을 수정할 수 있습니다.

배포에서 여러 로그 전달자를 사용할 때 ClusterLogging CR에 적용되는 스탠자는 managementState 및 collection입니다. 다른 모든 스탠자는 무시됩니다.

사전 요구 사항

-

관리자 권한이 있습니다.

-

Red Hat OpenShift Logging Operator 버전 5.8 이상이 설치되어 있습니다.

- ClusterLogForwarder CR을 생성했습니다.

절차

1.

기존 ClusterLogForwarder CR을 지원하는 ClusterLogging CR을 생성합니다.

ClusterLogging CR YAML의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
  name: <name> ①
  namespace: <namespace> ②
spec:
  managementState: "Managed"
  collection:
    type: "vector"
    tolerations:
      - key: "logging"
        operator: "Exists"
        effect: "NoExecute"
        tolerationSeconds: 6000
    resources:
      limits:
        memory: 1Gi
      requests:
        cpu: 100m
        memory: 1Gi
    nodeSelector:
      collector: needed
# ...
```

1

이름은 ClusterLogForwarder CR과 동일해야 합니다.

2

네임스페이스는 ClusterLogForwarder CR과 동일해야 합니다.

2.

다음 명령을 실행하여 ClusterLogging CR을 적용합니다.

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

13.1.4. 로깅 수집기 Pod 보기

로깅 수집기 Pod 및 실행 중인 해당 노드를 볼 수 있습니다.

절차

-

프로젝트에서 다음 명령을 실행하여 로깅 수집기 Pod 및 세부 정보를 확인합니다.

```
$ oc get pods --selector component=collector -o wide -n <project_name>
```

출력 예

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

13.1.5. 추가 리소스

-

[노드 선택기를 사용하여 특정 노드에 Pod 배치](#)

13.2. 테인트 및 허용 오차를 사용하여 로깅 POD 배치 제어

테인트 및 허용 오차를 사용하면 노드에서 예약해야 하는 (또는 예약해서는 안 되는) Pod를 제어할 수

있습니다.

13.2.1. 테인트(Taints) 및 톨러레이션(Tolerations)의 이해

테인트를 사용하면 Pod에 일치하는 허용 오차가 없는 경우 노드에서 Pod 예약을 거부할 수 있습니다.

Node 사양(NodeSpec)을 통해 노드에 테인트를 적용하고 **Pod 사양(PodSpec)**을 통해 Pod에 허용 오차를 적용합니다. 노드에 테인트를 적용할 때 Pod에서 테인트를 허용할 수 없는 경우 스케줄러에서 해당 노드에 Pod를 배치할 수 없습니다.

노드 사양의 테인트 예

```
apiVersion: v1
kind: Node
metadata:
  name: my-node
#...
spec:
  taints:
    - effect: NoExecute
      key: key1
      value: value1
#...
```

Pod 사양의 허용 오차 예

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
#...
spec:
  tolerations:
    - key: "key1"
      operator: "Equal"
      value: "value1"
      effect: "NoExecute"
      tolerationSeconds: 3600
#...
```

테인트 및 톨러레이션은 **key**, **value** 및 **effect**로 구성되어 있습니다.

표 13.1. 테인트 및 톨러레이션 구성 요소

매개변수	설명						
key	key 는 최대 253 자의 문자열입니다. 키는 문자 또는 숫자로 시작해야 하며 문자, 숫자, 하이픈, 점, 밑줄을 포함할 수 있습니다.						
value	value 는 최대 63 자의 문자열입니다. 값은 문자 또는 숫자로 시작해야 하며 문자, 숫자, 하이픈, 점, 밑줄을 포함할 수 있습니다.						
effect	다음 명령 중 하나를 실행합니다. <table border="1" style="margin-left: 20px;"> <tr> <td>NoSchedule^[1]</td><td> <ul style="list-style-type: none"> 테인트에 일치하지 않는 새 pod는 해당 노드에 예약되지 않습니다. 노드의 기존 pod는 그대로 유지됩니다. </td></tr> <tr> <td>PreferNoSchedule</td><td> <ul style="list-style-type: none"> 테인트와 일치하지 않는 새 pod는 해당 노드에 예약할 수 있지만 스케줄러는 그 렇게하지 않습니다. 노드의 기존 pod는 그대로 유지됩니다. </td></tr> <tr> <td>NoExecute</td><td> <ul style="list-style-type: none"> 테인트에 일치하지 않는 새 pod는 해당 노드에 예약할 수 없습니다. 일치하는 톨러레이션이 없는 노드의 기존 pod는 제거됩니다. </td></tr> </table>	NoSchedule ^[1]	<ul style="list-style-type: none"> 테인트에 일치하지 않는 새 pod는 해당 노드에 예약되지 않습니다. 노드의 기존 pod는 그대로 유지됩니다. 	PreferNoSchedule	<ul style="list-style-type: none"> 테인트와 일치하지 않는 새 pod는 해당 노드에 예약할 수 있지만 스케줄러는 그 렇게하지 않습니다. 노드의 기존 pod는 그대로 유지됩니다. 	NoExecute	<ul style="list-style-type: none"> 테인트에 일치하지 않는 새 pod는 해당 노드에 예약할 수 없습니다. 일치하는 톨러레이션이 없는 노드의 기존 pod는 제거됩니다.
NoSchedule ^[1]	<ul style="list-style-type: none"> 테인트에 일치하지 않는 새 pod는 해당 노드에 예약되지 않습니다. 노드의 기존 pod는 그대로 유지됩니다. 						
PreferNoSchedule	<ul style="list-style-type: none"> 테인트와 일치하지 않는 새 pod는 해당 노드에 예약할 수 있지만 스케줄러는 그 렇게하지 않습니다. 노드의 기존 pod는 그대로 유지됩니다. 						
NoExecute	<ul style="list-style-type: none"> 테인트에 일치하지 않는 새 pod는 해당 노드에 예약할 수 없습니다. 일치하는 톨러레이션이 없는 노드의 기존 pod는 제거됩니다. 						
operator	<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Equal</td><td>key/value/effect 매개변수가 일치해야 합니다. 이는 기본값입니다.</td></tr> <tr> <td style="width: 30%;">Exists</td><td>key/effect 매개변수가 일치해야 합니다. 일치하는 빈 value 매개변수를 남겨 두어야 합니다.</td></tr> </table>	Equal	key/value/effect 매개변수가 일치해야 합니다. 이는 기본값입니다.	Exists	key/effect 매개변수가 일치해야 합니다. 일치하는 빈 value 매개변수를 남겨 두어야 합니다.		
Equal	key/value/effect 매개변수가 일치해야 합니다. 이는 기본값입니다.						
Exists	key/effect 매개변수가 일치해야 합니다. 일치하는 빈 value 매개변수를 남겨 두어야 합니다.						

1.

컨트롤 플레인 노드에 **NoSchedule** 테인트를 추가하는 경우 노드에 기본적으로 추가되는 **node-role.kubernetes.io/master=:NoSchedule** 테인트가 있어야 합니다.

예를 들면 다음과 같습니다.

```
apiVersion: v1
kind: Node
metadata:
  annotations:
    machine.openshift.io/machine: openshift-machine-api/ci-ln-62s7gtb-f76d1-v8jxv-
    master-0
    machineconfiguration.openshift.io/currentConfig: rendered-master-
    cdc1ab7da414629332cc4c3926e6e59c
    name: my-node
  #...
spec:
  taints:
  - effect: NoSchedule
    key: node-role.kubernetes.io/master
  #...
```

톨러레이션은 테인트와 일치합니다.

- **operator** 매개변수가 **Equal**로 설정된 경우:
 - **key** 매개변수는 동일합니다.
 - **value** 매개변수는 동일합니다.
 - **effect** 매개변수는 동일합니다.
- **operator** 매개변수가 **Exists**로 설정된 경우:
 - **key** 매개변수는 동일합니다.
 - **effect** 매개변수는 동일합니다.

다음 테인트는 OpenShift Container Platform에 빌드됩니다.

- **node.kubernetes.io/not-ready**: 노드가 준비 상태에 있지 않습니다. 이는 노드 조건 Ready=False에 해당합니다.
- **node.kubernetes.io/unreachable**: 노드가 노드 컨트롤러에서 연결할 수 없습니다. 이는 노드 조건 Ready=Unknown에 해당합니다.
- **node.kubernetes.io/memory-pressure**: 노드에 메모리 부족 문제가 있습니다. 이는 노드 조건 MemoryPressure=True에 해당합니다.
- **node.kubernetes.io/disk-pressure**: 노드에 디스크 부족 문제가 있습니다. 이는 노드 조건 DiskPressure=True에 해당합니다.
- **node.kubernetes.io/network-unavailable**: 노드 네트워크를 사용할 수 없습니다.
- **node.kubernetes.io/unschedulable**: 노드를 예약할 수 없습니다.
- **node.cloudprovider.kubernetes.io/uninitialized**: 노드 컨트롤러가 외부 클라우드 공급자로 시작되면 이 테인트 노드에 사용 불가능으로 표시됩니다. cloud-controller-manager의 컨트롤러가 이 노드를 초기화하면 kubelet이 이 테인트를 제거합니다.
- **node.kubernetes.io/pid-pressure**: 노드에 pid pressure가 있습니다. 이는 노드 조건 PIDPressure=True에 해당합니다.



중요

OpenShift Container Platform은 기본 pid.available evictionHard 를 설정하지 않습니다.

13.2.2. Loki Pod 배치

Pod의 허용 오차 또는 노드 선택기를 사용하여 **Loki Pod**가 실행되는 노드를 제어하고 다른 워크로드가 해당 노드를 사용하지 못하도록 할 수 있습니다.

LokiStack CR(사용자 정의 리소스)을 사용하여 로그 저장소 **Pod**에 허용 오차를 적용하고 노드 사양이 있는 노드에 **taint**를 적용할 수 있습니다. 노드의 테인트는 테인트를 허용하지 않는 모든 **Pod**를 거절하도록 노드에 지시하는 키:값 쌍입니다. 다른 **Pod**에 없는 특정 키:값 쌍을 사용하면 해당 노드에서 로그 저장소 **Pod**만 실행할 수 있습니다.

노드 선택기가 있는 **LokiStack**의 예

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
# ...
  template:
    compactor: ①
      nodeSelector:
        node-role.kubernetes.io/infra: "" ②
    distributor:
      nodeSelector:
        node-role.kubernetes.io/infra: ""
    gateway:
      nodeSelector:
        node-role.kubernetes.io/infra: ""
    indexGateway:
      nodeSelector:
        node-role.kubernetes.io/infra: ""
    ingester:
      nodeSelector:
        node-role.kubernetes.io/infra: ""
    querier:
      nodeSelector:
        node-role.kubernetes.io/infra: ""
    queryFrontend:
      nodeSelector:
        node-role.kubernetes.io/infra: ""
    ruler:
      nodeSelector:
        node-role.kubernetes.io/infra: ""
# ...
```

2

정의된 라벨이 포함된 노드로 이동되는 Pod를 지정합니다.

이전 예제 구성에서 모든 Loki Pod는 node-role.kubernetes.io/infra: "" 라벨이 포함된 노드로 이동합니다.

노드 선택기 및 허용 오차가 있는 LokiStack CR의 예

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
# ...
  template:
    compactor:
      nodeSelector:
        node-role.kubernetes.io/infra: ""
      tolerations:
        - effect: NoSchedule
          key: node-role.kubernetes.io/infra
          value: reserved
        - effect: NoExecute
          key: node-role.kubernetes.io/infra
          value: reserved
    distributor:
      nodeSelector:
        node-role.kubernetes.io/infra: ""
      tolerations:
        - effect: NoSchedule
          key: node-role.kubernetes.io/infra
          value: reserved
        - effect: NoExecute
          key: node-role.kubernetes.io/infra
          value: reserved
      nodeSelector:
        node-role.kubernetes.io/infra: ""
      tolerations:
        - effect: NoSchedule
          key: node-role.kubernetes.io/infra
          value: reserved
        - effect: NoExecute
          key: node-role.kubernetes.io/infra
          value: reserved
    indexGateway:
```

```
nodeSelector:
  node-role.kubernetes.io/infra: ""
tolerations:
- effect: NoSchedule
  key: node-role.kubernetes.io/infra
  value: reserved
- effect: NoExecute
  key: node-role.kubernetes.io/infra
  value: reserved
ingester:
nodeSelector:
  node-role.kubernetes.io/infra: ""
tolerations:
- effect: NoSchedule
  key: node-role.kubernetes.io/infra
  value: reserved
- effect: NoExecute
  key: node-role.kubernetes.io/infra
  value: reserved
querier:
nodeSelector:
  node-role.kubernetes.io/infra: ""
tolerations:
- effect: NoSchedule
  key: node-role.kubernetes.io/infra
  value: reserved
- effect: NoExecute
  key: node-role.kubernetes.io/infra
  value: reserved
queryFrontend:
nodeSelector:
  node-role.kubernetes.io/infra: ""
tolerations:
- effect: NoSchedule
  key: node-role.kubernetes.io/infra
  value: reserved
- effect: NoExecute
  key: node-role.kubernetes.io/infra
  value: reserved
ruler:
nodeSelector:
  node-role.kubernetes.io/infra: ""
tolerations:
- effect: NoSchedule
  key: node-role.kubernetes.io/infra
  value: reserved
- effect: NoExecute
  key: node-role.kubernetes.io/infra
  value: reserved
gateway:
nodeSelector:
  node-role.kubernetes.io/infra: ""
tolerations:
- effect: NoSchedule
  key: node-role.kubernetes.io/infra
  value: reserved
```

```

- effect: NoExecute
  key: node-role.kubernetes.io/infra
  value: reserved
# ...

```

LokiStack(CR)의 **nodeSelector** 및 허용 오차 필드를 구성하려면 **oc explain** 명령을 사용하여 특정 리소스에 대한 설명 및 필드를 볼 수 있습니다.

```
$ oc explain lokistack.spec.template
```

출력 예

```

KIND: LokiStack
VERSION: loki.grafana.com/v1

RESOURCE: template <Object>

DESCRIPTION:
Template defines the resource/limits/tolerations/nodeselectors per
component

FIELDS:
compactor <Object>
Compactor defines the compaction component spec.

distributor <Object>
Distributor defines the distributor component spec.

...

```

자세한 내용은 특정 필드를 추가할 수 있습니다.

```
$ oc explain lokistack.spec.template.compactor
```

출력 예

```
KIND: LokiStack
```

VERSION: loki.grafana.com/v1

RESOURCE: compactor <Object>

DESCRIPTION:

Compactor defines the compaction component spec.

FIELDS:

nodeSelector <map[string]string>

NodeSelector defines the labels required by a node to schedule the component onto it.

...

13.2.3. 허용 오차를 사용하여 로그 수집기 Pod 배치 제어

기본적으로 로그 수집기 Pod에는 다음과 같은 허용 오차 구성이 있습니다.

```
apiVersion: v1
kind: Pod
metadata:
  name: collector-example
  namespace: openshift-logging
spec:
# ...
  collection:
    type: vector
    tolerations:
      - effect: NoSchedule
        key: node-role.kubernetes.io/master
        operator: Exists
      - effect: NoSchedule
        key: node.kubernetes.io/disk-pressure
        operator: Exists
      - effect: NoExecute
        key: node.kubernetes.io/not-ready
        operator: Exists
      - effect: NoExecute
        key: node.kubernetes.io/unreachable
        operator: Exists
      - effect: NoSchedule
        key: node.kubernetes.io/memory-pressure
        operator: Exists
      - effect: NoSchedule
        key: node.kubernetes.io/pid-pressure
        operator: Exists
      - effect: NoSchedule
        key: node.kubernetes.io/unschedulable
        operator: Exists
# ...
```

사전 요구 사항

- Red Hat OpenShift Logging Operator 및 OpenShift CLI(oc)를 설치했습니다.

절차

1.

다음 명령을 실행하여 로깅 수집기 Pod에서 로깅 수집기 Pod를 예약할 노드에 테인트를 추가합니다.

```
$ oc adm taint nodes <node_name> <key>=<value>:<effect>
```

명령 예

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

이 예에서는 키 **collector**, 값 **node** 및 **taint** 효과 **NoExecute**로 **node1**에 **taint**를 배치합니다. **NoExecute** 허용을 사용해야 합니다. **NoExecute**는 **taint**와 일치하는 Pod만 스케줄링하고 일치하지 않는 기존 Pod는 제거합니다.

2.

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

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
# ...
spec:
# ...
collection:
type: vector
tolerations:
- key: collector 1
operator: Exists 2
effect: NoExecute 3
tolerationSeconds: 6000 4
resources:
limits:
memory: 2Gi
```

```

| requests:
|   cpu: 100m
|   memory: 1Gi
|
# ...

```

1

노드에 추가한 키를 지정합니다.

2

key/value/effect 매개변수가 일치할 것을 요구하도록 **Exists Operator**을 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 Pod를 **node1**에 예약할 수 있습니다.

13.2.4. 로깅 수집기에 대한 리소스 구성 및 스케줄링

관리자는 동일한 네임스페이스에 있고 지원되는 **ClusterLogForwarder CR**과 동일한 이름이 있는 **ClusterLogging** 사용자 정의 리소스(CR)를 생성하여 수집기의 리소스 또는 스케줄링을 수정할 수 있습니다.

배포에서 여러 로그 전달자를 사용할 때 **ClusterLogging CR**에 적용되는 스텐자는 **managementState** 및 **collection**입니다. 다른 모든 스템자는 무시됩니다.

사전 요구 사항

- 관리자 권한이 있습니다.

- Red Hat OpenShift Logging Operator 버전 5.8 이상이 설치되어 있습니다.
- ClusterLogForwarder CR을 생성했습니다.

절차

- 기존 ClusterLogForwarder CR을 지원하는 ClusterLogging CR을 생성합니다.

ClusterLogging CR YAML의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
  name: <name> ①
  namespace: <namespace> ②
spec:
  managementState: "Managed"
  collection:
    type: "vector"
    tolerations:
      - key: "logging"
        operator: "Exists"
        effect: "NoExecute"
        tolerationSeconds: 6000
    resources:
      limits:
        memory: 1Gi
      requests:
        cpu: 100m
        memory: 1Gi
    nodeSelector:
      collector: needed
# ...
```

1

이름은 ClusterLogForwarder CR과 동일해야 합니다.

2

2.

다음 명령을 실행하여 ClusterLogging CR을 적용합니다.

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

13.2.5. 로깅 수집기 Pod 보기

로깅 수집기 Pod 및 실행 중인 해당 노드를 볼 수 있습니다.

절차

-

프로젝트에서 다음 명령을 실행하여 로깅 수집기 Pod 및 세부 정보를 확인합니다.

```
$ oc get pods --selector component=collector -o wide -n <project_name>
```

출력 예

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

13.2.6. 추가 리소스

-

노드 테인트를 사용하여 Pod 배치 제어

14장. 로깅 설치 제거

설치된 **Operator** 및 관련 **CR**(사용자 정의 리소스)을 제거하여 OpenShift Container Platform 클러스터에서 로깅을 제거할 수 있습니다.

14.1. 로깅 설치 제거

Red Hat OpenShift Logging Operator 및 ClusterLogging 사용자 정의 리소스(**CR**)를 삭제하여 로그 집계를 중지할 수 있습니다.

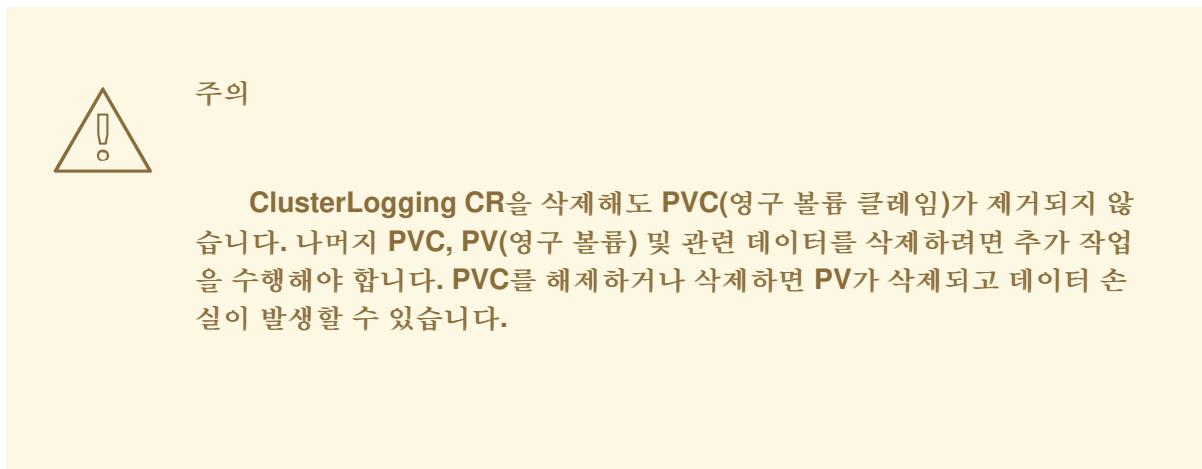
사전 요구 사항

- 관리자 권한이 있습니다.
- OpenShift Container Platform 웹 콘솔의 관리자 화면에 액세스할 수 있습니다.

절차

1. Administration → Custom Resource Definitions 페이지로 이동하여 ClusterLogging 을 클릭합니다.
2. 사용자 정의 리소스 정의 세부 정보 페이지에서 인스턴스를 클릭합니다.
3. 인스턴스 옆에 있는 옵션 메뉴  를 클릭하고 ClusterLogging 삭제 를 클릭합니다.
4. Administration → Custom Resource Definitions 페이지로 이동합니다.
5. ClusterLogging 옆에 있는 옵션 메뉴 

를 클릭하고 사용자 정의 리소스 정의 삭제를 선택합니다.



6.

ClusterLogForwarder CR을 생성한 경우 **ClusterLogForwarder** 옆에 있는 옵션 메뉴



를 클릭한 다음 사용자 정의 리소스 정의 삭제를 클릭합니다.

7.

Operator → 설치된 **Operator** 페이지로 이동합니다.

8.

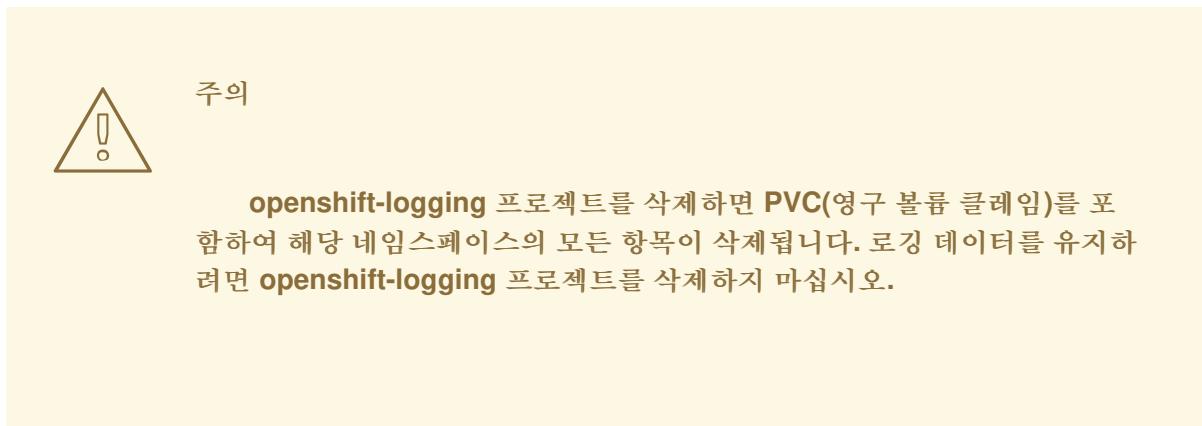
Red Hat OpenShift Logging Operator 옆에 있는 옵션 메뉴



를 클릭한 다음 **Operator** 설치 제거를 클릭합니다.

9.

선택 사항: **openshift-logging** 프로젝트를 삭제합니다.



a.
홈 → 프로젝트 페이지로 이동합니다.

b.
openshift-logging 프로젝트 옆에 있는 옵션 메뉴



를 클릭한 다음 프로젝트 삭제를 클릭합니다.

c.
대화 상자에 **openshift-logging**을 입력하여 삭제를 확인한 다음 삭제를 클릭합니다.

14.2. 로깅 PVC 삭제

다른 Pod와 재사용 할 수 있도록 PVC(영구 볼륨 클레임)를 유지하려면 PVC를 회수하는 데 필요한 레이블 또는 PVC 이름을 유지합니다. PVC를 유지하지 않으려면 해당 PVC를 삭제할 수 있습니다. 스토리지 공간을 복구하려면 PV(영구 볼륨)도 삭제할 수 있습니다.

사전 요구 사항

- 관리자 권한이 있습니다.
- OpenShift Container Platform 웹 콘솔의 관리자 화면에 액세스할 수 있습니다.

절차

1. 스토리지 → 영구 볼륨 클레임 페이지로 이동합니다.

2. 각 PVC 옆에 있는 옵션 메뉴



를 클릭하고 영구 볼륨 클레임 삭제를 선택합니다.

14.3. LOKI 설치 제거

사전 요구 사항

- 관리자 권한이 있습니다.
- **OpenShift Container Platform** 웹 콘솔의 관리자 화면에 액세스할 수 있습니다.
- **Red Hat OpenShift Logging Operator** 및 관련 리소스를 아직 제거하지 않은 경우 ClusterLogging 사용자 정의 리소스에서 **LokiStack**에 대한 참조를 제거했습니다.

절차

1. **Administration** → **Custom Resource Definitions** 페이지로 이동하여 **LokiStack** 을 클릭합니다.
2. 사용자 정의 리소스 정의 세부 정보 페이지에서 인스턴스를 클릭합니다.
3. 인스턴스 옆에 있는 옵션 메뉴  를 클릭한 다음 **LokiStack** 삭제를 클릭합니다.
4. **Administration** → **Custom Resource Definitions** 페이지로 이동합니다.
5. **LokiStack** 옆에 있는 옵션 메뉴  를 클릭하고 사용자 정의 리소스 정의 삭제를 선택합니다.
6. 오브젝트 스토리지 시크릿을 삭제합니다.
7. **Operator** → 설치된 **Operator** 페이지로 이동합니다.
8. **Loki Operator** 옆에 있는 옵션 메뉴



를 클릭한 다음 **Operator** 설치 제거를 클릭합니다.

9.

선택 사항: **openshift-operators-redhat** 프로젝트를 삭제합니다.



중요

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

a.

홈 → 프로젝트 페이지로 이동합니다.

b.

openshift-operators-redhat 프로젝트 옆에 있는 옵션 메뉴



를 클릭한 다음 프로젝트 삭제를 클릭합니다.

c.

대화 상자에 **openshift-operators-redhat** 을 입력하여 삭제를 확인한 다음 삭제를 클릭합니다.

14.4. ELASTICSEARCH 설치 제거

사전 요구 사항

- 관리자 권한이 있습니다.
- OpenShift Container Platform 웹 콘솔의 관리자 화면에 액세스할 수 있습니다.
- Red Hat OpenShift Logging Operator 및 관련 리소스를 아직 제거하지 않은 경우 ClusterLogging 사용자 정의 리소스에서 Elasticsearch에 대한 참조를 제거해야 합니다.

절차

1. 관리 → 사용자 정의 리소스 정의 페이지로 이동하여 **Elasticsearch** 를 클릭합니다.
2. 사용자 정의 리소스 정의 세부 정보 페이지에서 인스턴스를 클릭합니다.
3. 인스턴스 옆에 있는 옵션 메뉴 를 클릭한 다음 **Elasticsearch** 삭제 를 클릭합니다.
4. Administration → Custom Resource Definitions 페이지로 이동합니다.
5. Elasticsearch 옆에 있는 옵션 메뉴 를 클릭하고 사용자 정의 리소스 정의 삭제 를 선택합니다.
6. 오브젝트 스토리지 시크릿을 삭제합니다.
7. Operator → 설치된 Operator 페이지로 이동합니다.
8. OpenShift Elasticsearch Operator 옆에 있는 옵션 메뉴 를 클릭한 다음 Operator 설치 제거를 클릭합니다.
9. 선택 사항: openshift-operators-redhat 프로젝트를 삭제합니다.



중요

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

a.

홈 → 프로젝트 페이지로 이동합니다.

b.

openshift-operators-redhat 프로젝트 옆에 있는 옵션 메뉴



를 클릭한 다음 프로젝트 삭제 를 클릭합니다.

c.

대화 상자에 **openshift-operators-redhat** 을 입력하여 삭제를 확인한 다음 삭제 를 클릭 합니다.

14.5. CLI를 사용하여 클러스터에서 OPERATOR 삭제

클러스터 관리자는 **CLI**를 사용하여 선택한 네임스페이스에서 설치된 **Operator**를 삭제할 수 있습니다.

사전 요구 사항

- cluster-admin 권한이 있는 계정을 사용하여 OpenShift Container Platform 클러스터에 액세스할 수 있습니다.
- oc 명령이 워크스테이션에 설치되어 있습니다.

절차

1.

구독된 Operator의 최신 버전(예: 서비스-operator)이 currentCSV 필드에서 식별되는지 확인합니다.

```
$ oc get subscription.operators.coreos.com serverless-operator -n openshift-serverless -o yaml | grep currentCSV
```

출력 예

```
currentCSV: serverless-operator.v1.28.0
```

2.

서브스크립션을 삭제합니다(예: 서비스-operator).

```
$ oc delete subscription.operators.coreos.com serverless-operator -n openshift-serverless
```

출력 예

```
subscription.operators.coreos.com "serverless-operator" deleted
```

3.

이전 단계의 currentCSV 값을 사용하여 대상 네임스페이스에서 Operator의 CSV를 삭제합니다.

```
$ oc delete clusterserviceversion serverless-operator.v1.28.0 -n openshift-serverless
```

출력 예

```
clusterserviceversion.operators.coreos.com "serverless-operator.v1.28.0" deleted
```

추가 리소스

-

[수동으로 영구 볼륨 회수](#)

15장. 로그 레코드 필드

로깅에서 내보낸 로그 레코드에 다음 필드가 있을 수 있습니다. 로그 레코드는 일반적으로 **JSON** 개체로 포맷되지만 동일한 데이터 모델을 다른 인코딩에 적용할 수 있습니다.

Elasticsearch 및 **Kibana**에서 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.

MESSAGE

원본 로그 항목 텍스트 **UTF-8**로 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	HAPPY

STRUCTURED

구조화된 오브젝트인 원본 로그 항목입니다. 이 필드는 **Forwarder**가 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 제한이 없습니다.

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

@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 **Elasticsearch**를 사용하여 "@timestamp"를 찾습니다.

데이터 유형	date
예시 값	2015-01-24 14:06:05.071000000 Z

호스트 이름

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

데이터 유형	keyword
--------	---------

IPADDR4

소스 서버의 **IPv4** 주소입니다. 배열이 될 수 있습니다.

데이터 유형	ip
--------	----

IPADDR6

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

데이터 유형	ip
--------	----

LEVEL

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

다음 값은 [syslog.h](#)에서 가져오고 그 앞에 **해당하는 숫자**가 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** 로깅에서는 **CRITICAL**은 **crit**로 **ERROR**은 **err** 등과 일치시킬 수 있습니다.

데이터 유형	keyword
예시 값	info

PID

사용 가능한 경우 이는 로깅 엔티티의 프로세스 ID입니다.

데이터 유형	keyword
--------	---------

SERVICE

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

데이터 유형	keyword
--------	---------

16장. TAGS

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

데이터 유형	text
--------	------

FILE

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

데이터 유형	text
--------	------

OFFSET

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

데이터 유형	long
--------	------

17장. KUBERNETES

쿠버네티스 관련 메타데이터의 네임스페이스입니다.

데이터 유형	group
--------	-------

17.1. KUBERNETES.POD_NAME

Pod의 이름입니다.

데이터 유형	keyword
--------	---------

17.2. KUBERNETES.POD_ID

Pod의 Kubernetes ID입니다.

데이터 유형	keyword
--------	---------

17.3. KUBERNETES.NAMESPACE_NAME

Kubernetes의 네임스페이스 이름입니다.

데이터 유형	keyword
--------	---------

17.4. KUBERNETES.NAMESPACE_ID

Kubernetes의 네임스페이스 ID입니다.

데이터 유형	keyword
--------	---------

17.5. KUBERNETES.HOST

Kubernetes 노드 이름입니다.

데이터 유형	keyword
--------	---------

17.6. KUBERNETES.CONTAINER_NAME

Kubernetes의 컨테이너 이름입니다.

데이터 유형	keyword
--------	---------

17.7. KUBERNETES.ANNOTATIONS

Kubernetes 오브젝트와 관련된 주석입니다.

데이터 유형	group
--------	-------

17.8. KUBERNETES.LABELS

원래 **Kubernetes Pod**에 있는 레이블입니다.

데이터 유형	group
--------	-------

17.9. KUBERNETES.EVENT

Kubernetes 마스터 API에서 얻은 **Kubernetes** 이벤트입니다. 이 이벤트 설명은 [Event v1 코어의 유형](#) 이벤트를 느슨하게 따릅니다.

데이터 유형	group
--------	-------

17.9.1. kubernetes.event.verb

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

데이터 유형	keyword
예시 값	ADDED

17.9.2. kubernetes.event.metadata

이벤트 생성 위치 및 시간 관련 정보입니다.

데이터 유형	group
--------	-------

17.9.2.1. kubernetes.event.metadata.name

이벤트 생성을 트리거한 오브젝트의 이름입니다.

데이터 유형	keyword
예시 값	java-mainclass-1.14d888a4cf24890

17.9.2.2. kubernetes.event.metadata.namespace

이벤트가 처음 발생한 네임스페이스의 이름입니다. **eventrouter** 애플리케이션이 배포된 네임스페이스인 **kubernetes.namespace_name**과 다릅니다.

데이터 유형	keyword
예시 값	default

17.9.2.3. kubernetes.event.metadata.selfLink

이벤트에 대한 링크입니다.

데이터 유형	keyword
예시 값	/api/v1/namespaces/javaj/events/java-mainclass-1.14d888a4cf24890

17.9.2.4. kubernetes.event.metadata.uid

이벤트의 고유 ID입니다.

데이터 유형	keyword
예시 값	d828ac69-7b58-11e7-9cf5-5254002f560c

17.9.2.5. kubernetes.event.metadata.resourceVersion

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

데이터 유형	integer
예시 값	311987

17.9.3. kubernetes.event.involvedObject

이벤트의 오브젝트입니다.

데이터 유형	group
--------	-------

17.9.3.1. kubernetes.event.involvedObject.kind

오브젝트 유형입니다.

데이터 유형	keyword
예시 값	ReplicationController

17.9.3.2. kubernetes.event.involvedObject.namespace

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

데이터 유형	keyword
--------	---------

예시 값	default
------	----------------

17.9.3.3. kubernetes.event.involvedObject.name

이벤트를 트리거한 오브젝트의 이름입니다.

데이터 유형	keyword
예시 값	java-mainclass-1

17.9.3.4. kubernetes.event.involvedObject.uid

오브젝트의 고유 ID입니다.

데이터 유형	keyword
예시 값	e6bff941-76a8-11e7-8193-5254002f560c

17.9.3.5. kubernetes.event.involvedObject.apiVersion

kubernetes 마스터 API의 버전입니다.

데이터 유형	keyword
예시 값	v1

17.9.3.6. 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	308882

17.9.4. kubernetes.event.reason

이 이벤트를 생성하는 이유를 제공하는 짧은 며신 이해 문자열입니다.

데이터 유형	keyword
예시 값	SuccessfulCreate

17.9.5. kubernetes.event.source_component

이 이벤트를 보고한 구성 요소입니다.

데이터 유형	keyword
예시 값	replication-controller

17.9.6. kubernetes.event.firstTimestamp

이벤트가 처음 기록된 시간입니다.

데이터 유형	date
예시 값	2017-08-07 10:11:57.000000000 Z

17.9.7. kubernetes.event.count

이 이벤트가 발생한 횟수입니다.

데이터 유형	integer
예시 값	1

17.9.8. kubernetes.event.type

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

데이터 유형	keyword
--------	---------

예시 값	Normal
------	---------------

18장. OPENSHIFT

openshift-logging 특정 메타데이터의 네임스페이스

데이터 유형	group
--------	-------

18.1. OPENSHIFT.LABELS

Cluster Log Forwarder 구성에 추가된 레이블

데이터 유형	group
--------	-------

19장. API 참조

19.1. 5.6 로깅 API 참조

19.1.1. 로깅 5.6 API 참조

19.1.1.1. ClusterLogForwarder

ClusterLogForwarder는 로그 전달을 구성하는 API입니다.

이름 지정된 입력 집합에서 이름이 지정된 출력 세트로 전달하는 파이프라인 목록을 지정하여 전달을 구성합니다.

공통 로그 카테고리에 대한 기본 제공 입력 이름이 있으며 사용자 지정 입력을 정의하여 추가 필터링을 수행할 수 있습니다.

기본 **openshift** 로그 저장소에는 기본 제공 출력 이름이 있지만 **URL** 및 기타 연결 정보로 자체 출력을 정의하여 클러스터 내부 또는 외부의 다른 저장소 또는 프로세서로 로그를 전달할 수 있습니다.

자세한 내용은 **API 필드**에 대한 설명서를 참조하십시오.

속성	유형	설명
spec	object	ClusterLogForwarder의 원하는 동작 사양
status	object	ClusterLogForwarder의 상태

19.1.1.1.1. .spec

19.1.1.1.1.1. 설명

ClusterLogForwarderSpec은 로그를 원격 대상으로 전달하는 방법을 정의합니다.

19.1.1.1.1.1.1. 유형

-

object

속성	유형	설명
입력	array	(선택 사항) 입력은 로그 메시지를 전달할 때 필터로 지정됩니다.
outputDefaults	object	(선택 사항) DEPRECATED OutputDefaults는 기본 저장소에 명시적으로 전달자 구성을 지정합니다.
출력	array	(선택 사항) 출력에는 로그 메시지에 대한 대상의 이름이 지정됩니다.
Pipeline	array	파이프라인은 입력 집합에서 선택한 메시지를 출력 세트로 전달합니다.

19.1.1.1.2. .spec.inputs[]

19.1.1.1.2.1. 설명

InputSpec은 로그 메시지의 선택기를 정의합니다.

19.1.1.1.2.1.1. 유형

-

array

속성	유형	설명
애플리케이션	object	(선택 사항) 애플리케이션이 있는 경우 이름이 지정된 애플리케이션 로그 세트를 활성화합니다.
name	string	파이프라인의 입력을 참조하는 데 사용되는 이름입니다.

19.1.1.1.3. .spec.inputs[].application

19.1.1.1.3.1. 설명

애플리케이션 로그 선택기. 선택기의 모든 조건을 충족 (**logical AND**)하여 로그를 선택해야 합니다.

19.1.1.3.1.1. 유형

-

object

속성	유형	설명
네임스페이스	array	(선택 사항) 애플리케이션 로그를 수집 할 네임스페이스입니다.
선택기	object	(선택 사항) 일치하는 라벨이 있는 Pod의 로그 선택기입니다.

19.1.1.4. .spec.inputs[].application.namespaces[]

19.1.1.4.1. 설명

19.1.1.4.1.1. 유형

-

array

19.1.1.5. .spec.inputs[].application.selector

19.1.1.5.1. 설명

레이블 선택기는 리소스 집합에 대한 레이블 쿼리입니다.

19.1.1.5.1.1. 유형

-

object

속성	유형	설명
matchLabels	object	(선택 사항) matchLabels는 {key,value} 쌍의 맵입니다. matchLabels의 단일 {key,value}

19.1.1.1.6. .spec.inputs[].application.selector.matchLabels**19.1.1.1.6.1. 설명****19.1.1.1.6.1.1. 유형**

- **object**

19.1.1.1.7. .spec.outputDefaults**19.1.1.1.7.1. 설명****19.1.1.1.7.1.1. 유형**

- **object**

속성	유형	설명
elasticsearch	object	(선택 사항) Elasticsearch OutputSpec 기본값

19.1.1.1.8. .spec.outputDefaults.elasticsearch**19.1.1.1.8.1. 설명**

ElasticsearchStructuredSpec은 **elasticsearch** 인덱스를 결정하기 위해 구조화된 로그 변경과 관련된 사양입니다.

19.1.1.1.8.1.1. 유형

- **object**

속성	유형	설명
enableStructuredContainerLogs	bool	(선택 사항) EnableStructuredContainerLogs를 사용하면 다중 컨테이너 구조화된 로그를 허용할 수 있습니다.

속성	유형	설명
structuredTypeKey	string	(선택 사항) elasticsearch 인덱스의 이름으로 사용할 메타데이터 키를 지정합니다.
structuredTypeName	string	(선택 사항) loadbalancingdTypeName은 elasticsearch 스키마의 이름을 지정합니다.

19.1.1.9. .spec.outputs[]

19.1.1.9.1. 설명

출력은 로그 메시지의 대상을 정의합니다.

19.1.1.9.1.1. 유형

-

array

속성	유형	설명
syslog	object	(선택 사항)
fluentdForward	object	(선택 사항)
elasticsearch	object	(선택 사항)
kafka	object	(선택 사항)
cloudwatch	object	(선택 사항)
loki	object	(선택 사항)
googleCloudLogging	object	(선택 사항)
splunk	object	(선택 사항)
name	string	파이프라인의 출력을 참조하는 데 사용되는 이름입니다.
secret	object	(선택 사항) 인증을 위한 시크릿입니다.

속성	유형	설명
tls	object	TLS에는 TLS 클라이언트 연결에 대한 옵션을 제어하는 설정이 포함되어 있습니다.
type	string	출력 플러그인의 유형입니다.
url	string	(선택 사항) 로그 레코드를 보낼 URL입니다.

19.1.1.1.10. .spec.outputs[].secret

19.1.1.1.10.1. 설명

OutputSecretSpec은 네임스페이스가 없는 이름만 포함하는 보안 참조입니다.

19.1.1.1.10.1.1. 유형

- **object**

속성	유형	설명
name	string	로그 전달자 보안에 대해 구성된 네임스페이스의 보안 이름입니다.

19.1.1.1.11. .spec.outputs[].tls

19.1.1.1.11.1. 설명

OutputTLSSpec에는 출력 유형과 무관한 TLS 연결 옵션이 포함되어 있습니다.

19.1.1.1.11.1.1. 유형

- **object**

속성	유형	설명

속성	유형	설명
insecureSkipVerify	bool	InsecureSkipVerify이 true인 경우 인증서의 오류를 무시하도록 TLS 클라이언트가 구성됩니다.

19.1.1.12. .spec.pipelines[]

19.1.1.12.1. 설명

PipelinesSpec은 입력 세트를 출력 세트에 연결합니다.

19.1.1.12.1.1. 유형

- **array**

속성	유형	설명
detectMultilineErrors	bool	(선택 사항) DetectMultilineErrors를 사용하면 컨테이너 로그를 여러 줄 오류 감지할 수 있습니다.
inputRefs	array	inputRefs는 이 파이프라인에 대한 입력 이름(input.name)을 나열합니다.
labels	object	(선택 사항) 이 파이프라인을 통과하는 로그 레코드에 적용되는 라벨입니다.
name	string	(선택 사항) 이름은 선택 사항이지만 파이프라인 목록에서 고유해야 합니다.
outputRefs	array	outputRefs는 이 파이프라인의 출력 이름(output.name)을 나열합니다.
구문 분석	string	(선택 사항) Parse를 사용하면 로그 항목을 구조화된 로그로 구문 분석 할 수 있습니다.

19.1.1.13. .spec.pipelines[].inputRefs[]

19.1.1.1.13.1. 설명**19.1.1.1.13.1.1. 유형**

- **array**

19.1.1.1.14. .spec.pipelines[].labels**19.1.1.1.14.1. 설명****19.1.1.1.14.1.1. 유형**

- **object**

19.1.1.1.15. .spec.pipelines[].outputRefs[]**19.1.1.1.15.1. 설명****19.1.1.1.15.1.1. 유형**

- **array**

19.1.1.1.16. .status**19.1.1.1.16.1. 설명**

ClusterLogForwarderStatus는 **ClusterLogForwarder**의 관찰 상태를 정의합니다.

19.1.1.1.16.1.1. 유형

- **object**

속성	유형	설명
conditions	object	로그 전달자의 상태입니다.
입력	조건	입력은 입력 이름을 입력 조건에 매핑합니다.

속성	유형	설명
출력	조건	출력은 출력 이름을 출력 조건에 매핑합니다.
Pipeline	조건	파이프라인은 파이프라인 이름을 파이프라인의 조건에 매핑합니다.

19.1.1.1.17. `.status.conditions`

19.1.1.1.17.1. 설명

19.1.1.1.17.1.1. 유형

- **object**

19.1.1.1.18. `.status.inputs`

19.1.1.1.18.1. 설명

19.1.1.1.18.1.1. 유형

- 조건

19.1.1.1.19. `.status.outputs`

19.1.1.1.19.1. 설명

19.1.1.1.19.1.1. 유형

- 조건

19.1.1.1.20. `.status.pipelines`

19.1.1.1.20.1. 설명

19.1.1.1.20.1.1. 유형

- **conditions== ClusterLogging A Red Hat OpenShift Logging 인스턴스.**
ClusterLogging은 clusterloggings API의 스키마입니다.

속성	유형	설명
spec	object	ClusterLogging의 원하는 동작 사양
status	object	status는 ClusterLogging의 관찰 상태를 정의합니다.

19.1.1.21. .spec

19.1.1.21.1. 설명

ClusterLoggingSpec은 원하는 **ClusterLogging** 상태를 정의합니다.

19.1.1.21.1.1. 유형

-

object

속성	유형	설명
컬렉션	object	클러스터의 컬렉션 구성 요소 사양
큐레이션	object	(DEPRECATED) (선택 사항) 더 이상 사용되지 않습니다. 클러스터의 Curation 구성 요소 사양
forwarder	object	(DEPRECATED) (선택 사항) 더 이상 사용되지 않습니다. 클러스터의 전달자 구성 요소 사양
logStore	object	(선택 사항) 클러스터의 로그 스토리지 구성 요소 사양
managementState	string	(선택 사항) 리소스가 운영자에 의해 '관리'인지 또는 '관리되지 않음'인지 나타냅니다.
시각화	object	(선택 사항) 클러스터에 대한 시각화 구성 요소의 사양

19.1.1.22. .spec.collection

19.1.1.22.1. 설명

이는 **Log** 및 이벤트 컬렉션에 대한 정보를 포함하는 구조입니다.

19.1.1.22.1.1. 유형

-

object

속성	유형	설명
resources	object	(선택 사항) 수집기의 리소스 요구 사항입니다.
nodeSelector	object	(선택 사항) Pod가 예약된 노드를 정의합니다.
허용 오차	array	(선택 사항) Pod에서 허용할 허용 오차를 정의합니다.
fluentd	object	(선택 사항) Fluentd는 fluentd 유형의 전달자 구성을 나타냅니다.
logs	object	(DEPRECATED)(선택 사항) 더 이상 사용되지 않습니다. 클러스터의 로그 컬렉션 사양
type	string	(선택 사항) 구성할 로그 컬렉션의 유형입니다.

19.1.1.23. .spec.collection.fluentd

19.1.1.23.1. 설명

FluentdForwarderSpec은 fluentd 유형의 전달자를 위한 구성을 나타냅니다.

19.1.1.23.1.1. 유형

-

object

속성	유형	설명
버퍼	object	
inFile	object	

19.1.1.1.24. .spec.collection.fluentd.buffer

19.1.1.1.24.1. 설명

FluentdBufferSpec은 모든 **fluentd** 출력에 대해 버퍼 구성을 튜닝하는 **fluentd** 버퍼 매개변수의 서브 세트를 나타냅니다. 버퍼 및 큐 크기 조정, 플러시 작업 및 플러시 플러시를 구성하는 매개변수 서브 세트를 지원합니다.

일반적인 매개변수는 <https://docs.fluentd.org/configuration/buffer-section#buffering-parameters>에서 참조하십시오.

플러시 매개변수의 경우 <https://docs.fluentd.org/configuration/buffer-section#flushing-parameters>를 참조하십시오.

재시도 매개변수는 <https://docs.fluentd.org/configuration/buffer-section#retries-parameters>를 참조하십시오.

19.1.1.1.24.1.1. 유형

-

object

속성	유형	설명
chunkLimitSize	string	(선택 사항) ChunkLimitSize는 각 청크의 최대 크기를 나타냅니다. 이 벤트 예 정
flushInterval	string	(선택 사항) FlushInterval은 두 개의 연속 플러시 사이에서 대기하는 시간 기간을 나타냅니다.

속성	유형	설명
flushMode	string	(선택 사항) FlushMode는 청크를 작성할 플러시 스레드의 모드를 나타냅니다. mode
flushThreadCount	int	(선택 사항) FlushThreadCount reprents the number of threads used by the fluentd buffer
overflowAction	string	(선택 사항) OverflowAction은 fluentd 버퍼 플러그인에 대한 작업을 나타냅니다.
retryMaxInterval	string	(선택 사항) RetryMaxInterval은 지수 백오프의 최대 시간 간격을 나타냅니다.
retryTimeout	string	(선택 사항) RetryTimeout은 포기하기 전에 재시도를 시도할 수 있는 최대 시간 간격을 나타냅니다.
retryType	string	(선택 사항) RetryType은 플러시 작업 재시도 유형을 나타냅니다. 플러시 작업 가능
retryWait	string	(선택 사항) RetryWait은 플러시를 위한 두 번 재시도 사이의 시간 기간을 나타냅니다.
totalLimitSize	string	(선택 사항) TotalLimitSize는 fluentd 당 허용된 노드 공간의 임계값을 나타냅니다.

19.1.1.25. .spec.collection.fluentd.inFile

19.1.1.25.1. 설명

FluentdInFileSpec은 **fluentd in-tail** 플러그인 매개변수의 서브 세트를 사용하여 모든 **fluentd in-tail** 입력에 대한 구성을 조정합니다.

일반적인 매개변수는 <https://docs.fluentd.org/input/tail#parameters>에서 참조하십시오.

19.1.1.25.1.1. 유형

-

object

속성	유형	설명
readLinesLimit	int	(선택 사항) ReadLinesLimit는 각 I/O 작업에서 읽을 행 수를 나타냅니다.

19.1.1.1.26. .spec.collection.logs

19.1.1.1.26.1. 설명

19.1.1.1.26.1.1. 유형

-

object

속성	유형	설명
fluentd	object	Fluentd 로그 수집 구성 요소의 사양
type	string	구성할 로그 컬렉션의 유형입니다.

19.1.1.1.27. .spec.collection.logs.fluentd

19.1.1.1.27.1. 설명

CollectorSpec은 수집기의 스케줄링 및 리소스를 정의하는 **spec**입니다.

19.1.1.1.27.1.1. 유형

-

object

속성	유형	설명
nodeSelector	object	(선택 사항) Pod가 예약된 노드를 정의합니다.

속성	유형	설명
resources	object	(선택 사항) 수집기의 리소스 요구 사항입니다.
허용 오차	array	(선택 사항) Pod에서 허용할 허용 오차를 정의합니다.

19.1.1.1.28. .spec.collection.logs.fluentd.nodeSelector

19.1.1.1.28.1. 설명

19.1.1.1.28.1.1. 유형

- **object**

19.1.1.1.29. .spec.collection.logs.fluentd.resources

19.1.1.1.29.1. 설명

19.1.1.1.29.1.1. 유형

- **object**

속성	유형	설명
limits	object	(선택 사항) 제한은 허용되는 최대 컴퓨팅 리소스 양을 설명합니다.
requests	object	(선택 사항) Requests는 필요한 최소 컴퓨팅 리소스 양을 설명합니다.

19.1.1.1.30. .spec.collection.logs.fluentd.resources.limits

19.1.1.1.30.1. 설명

19.1.1.1.30.1.1. 유형

- **object**

19.1.1.1.31. .spec.collection.logs.fluentd.resources.requests

19.1.1.31.1. 설명

19.1.1.31.1.1. 유형

-

object

19.1.1.32. .spec.collection.logs.fluentd.tolerations[]

19.1.1.32.1. 설명

19.1.1.32.1.1. 유형

-

array

속성	유형	설명
effect	string	(선택 사항) Effect는 일치하는 테인트 효과를 나타냅니다. 빈은 모든 테인트 효과와 일치함을 의미합니다.
key	string	(선택 사항) Key는 허용 오차가 적용되는 테인트 키입니다. 빈은 모든 테인트 키와 일치함을 의미합니다.
operator	string	(선택 사항) Operator는 값과 키의 관계를 나타냅니다.
tolerationSeconds	int	(선택 사항) TolerationSeconds는 허용 오차(선택 사항)를 나타냅니다.
value	string	(선택 사항) 값은 �oller레이션이 일치하는 테인트 값입니다.

19.1.1.33. .spec.collection.logs.fluentd.tolerations[].tolerationSeconds

19.1.1.33.1. 설명

19.1.1.33.1.1. 유형

-

int

19.1.1.1.34. .spec.curation

19.1.1.1.34.1. 설명

이는 로그 큐레이션(Curator)에 대한 정보를 포함하는 구조입니다.

19.1.1.1.34.1.1. 유형

-

object

속성	유형	설명
Curator	object	구성할 큐레이션의 사양
type	string	구성할 큐레이션의 종류

19.1.1.1.35. .spec.curation.curator

19.1.1.1.35.1. 설명

19.1.1.1.35.1.1. 유형

-

object

속성	유형	설명
nodeSelector	object	Pod가 예약된 노드를 정의합니다.
resources	object	(선택 사항) Curator의 리소스 요구 사항입니다.
스케줄	string	Curator 작업이 실행되는 cron 스케줄입니다. 기본값은 "30 3 * *"입니다.
허용 오차	array	

19.1.1.1.36. .spec.curation.curator.nodeSelector

19.1.1.1.36.1. 설명

19.1.1.1.36.1.1. 유형

- **object**

19.1.1.1.37. .spec.curation.curator.resources**19.1.1.1.37.1. 설명****19.1.1.1.37.1.1. 유형**

- **object**

속성	유형	설명
limits	object	(선택 사항) 제한은 허용되는 최대 컴퓨팅 리소스 양을 설명합니다.
requests	object	(선택 사항) Requests는 필요한 최소 컴퓨팅 리소스 양을 설명합니다.

19.1.1.1.38. .spec.curation.curator.resources.limits**19.1.1.1.38.1. 설명****19.1.1.1.38.1.1. 유형**

- **object**

19.1.1.1.39. .spec.curation.curator.resources.requests**19.1.1.1.39.1. 설명****19.1.1.1.39.1.1. 유형**

- **object**

19.1.1.1.40. .spec.curation.curator.tolerations[]**19.1.1.1.40.1. 설명**

19.1.1.40.1.1. 유형

-

array

속성	유형	설명
effect	string	(선택 사항) Effect는 일치하는 테인트 효과를 나타냅니다. 빈은 모든 테인트 효과와 일치함을 의미합니다.
key	string	(선택 사항) Key는 허용 오차가 적용되는 테인트 키입니다. 빈은 모든 테인트 키와 일치함을 의미합니다.
operator	string	(선택 사항) Operator는 값과 키의 관계를 나타냅니다.
tolerationSeconds	int	(선택 사항) TolerationSeconds는 허용 오차(선택 사항)를 나타냅니다.
value	string	(선택 사항) 값은 톨러레이션이 일치하는 테인트 값입니다.

19.1.1.41. .spec.curation.curator.tolerations[].tolerationSeconds

19.1.1.41.1. 설명

19.1.1.41.1.1. 유형

-

int

19.1.1.42. .spec.forwarder

19.1.1.42.1. 설명

ForwarderSpec에는 특정 전달자 구현을 위한 글로벌 튜닝 매개변수가 포함되어 있습니다. 이 필드는 일반적으로 사용할 필요가 없으며 기본 전달자 기술에 익숙한 사용자가 성능 튜닝을 허용합니다. 현재 지원됨: fluentd.

19.1.1.42.1.1. 유형

-

object

속성	유형	설명
fluentd	object	

19.1.1.1.43. .spec.forwarder.fluentd

19.1.1.1.43.1. 설명

FluentdForwarderSpec은 fluentd 유형의 전달자를 위한 구성을 나타냅니다.

19.1.1.1.43.1.1. 유형

-

object

속성	유형	설명
버퍼	object	
inFile	object	

19.1.1.1.44. .spec.forwarder.fluentd.buffer

19.1.1.1.44.1. 설명

FluentdBufferSpec은 모든 fluentd 출력에 대해 버퍼 구성을 튜닝하는 fluentd 버퍼 매개변수의 서브 세트를 나타냅니다. 버퍼 및 큐 크기 조정, 플러시 작업 및 플러시 플러시를 구성하는 매개변수 서브 세트를 지원합니다.

일반적인 매개변수는 <https://docs.fluentd.org/configuration/buffer-section#buffering-parameters>에서 참조하십시오.

플러시 매개변수의 경우 <https://docs.fluentd.org/configuration/buffer-section#flushing-parameters>를 참조하십시오.

재시도 매개변수는 <https://docs.fluentd.org/configuration/buffer-section#retries-parameters>를 참조하십시오.

19.1.1.44.1.1. 유형



object

속성	유형	설명
chunkLimitSize	string	(선택 사항) ChunkLimitSize는 각 청크의 최대 크기를 나타냅니다. 이 벤트 예정
flushInterval	string	(선택 사항) FlushInterval은 두 개의 연속 플러시 사이에서 대기하는 시간 기간을 나타냅니다.
flushMode	string	(선택 사항) FlushMode는 청크를 작성할 플러시 스레드의 모드를 나타냅니다. mode
flushThreadCount	int	(선택 사항) FlushThreadCount reprents the number of threads used by the fluentd buffer
overflowAction	string	(선택 사항) OverflowAction은 fluentd 버퍼 플러그인에 대한 작업을 나타냅니다.
retryMaxInterval	string	(선택 사항) RetryMaxInterval은 지수 백오프의 최대 시간 간격을 나타냅니다.
retryTimeout	string	(선택 사항) RetryTimeout은 포기하기 전에 재시도를 시도할 수 있는 최대 시간 간격을 나타냅니다.
retryType	string	(선택 사항) RetryType은 플러시 작업 재시도 유형을 나타냅니다. 플러시 작업 가능
retryWait	string	(선택 사항) RetryWait은 플러시를 위한 두 번 재시도 사이의 시간 기간을 나타냅니다.
totalLimitSize	string	(선택 사항) TotalLimitSize는 fluentd 당 허용된 노드 공간의 임계값을 나타냅니다.

19.1.1.1.45. .spec.forwarder.fluentd.inFile

19.1.1.1.45.1. 설명

FluentdInFileSpec은 **fluentd in-tail** 플러그인 매개변수의 서브 세트를 사용하여 모든 **fluentd in-tail** 입력에 대한 구성을 조정합니다.

일반적인 매개변수는 <https://docs.fluentd.org/input/tail#parameters>에서 참조하십시오.

19.1.1.1.45.1.1. 유형

-

object

속성	유형	설명
readLinesLimit	int	(선택 사항) ReadLinesLimit는 각 I/O 작업에서 읽을 행 수를 나타냅니다.

19.1.1.1.46. .spec.logStore

19.1.1.1.46.1. 설명

LogStoreSpec에는 로그 저장 방법에 대한 정보가 포함되어 있습니다.

19.1.1.1.46.1.1. 유형

-

object

속성	유형	설명
elasticsearch	object	Elasticsearch 로그 저장소 구성 요소의 사양
lokistack	object	CloudEventStack에는 Type이 LogStoreTypeLokiStack으로 설정된 경우 로그 스토리지에 사용할 ScanSettingStack에 대한 정보가 포함되어 있습니다.

속성	유형	설명
retentionPolicy	object	(선택 사항) 보존 정책은 인덱스를 삭제해야 하는 인덱스의 최대 기간을 정의합니다.
type	string	구성 할 로그 스토리지 유형입니다. Operator는 현재 Elasticsearch를 사용하여 지원

19.1.1.47. .spec.logStore.elasticsearch

19.1.1.47.1. 설명

19.1.1.47.1.1. 유형

- **object**

속성	유형	설명
nodeCount	int	Elasticsearch에 배포할 노드 수
nodeSelector	object	Pod가 예약된 노드를 정의합니다.
proxy	object	Elasticsearch 프록시 구성 요소의 사양
redundancyPolicy	string	(선택 사항)
resources	object	(선택 사항) Elasticsearch의 리소스 요구 사항입니다.
storage	object	(선택 사항) Elasticsearch 데이터 노드의 스토리지 사양
허용 오차	array	

19.1.1.48. .spec.logStore.elasticsearch.nodeSelector

19.1.1.48.1. 설명

19.1.1.48.1.1. 유형

-

object

19.1.1.1.49. .spec.logStore.elasticsearch.proxy

19.1.1.1.49.1. 설명

19.1.1.1.49.1.1. 유형

-

object

속성	유형	설명
resources	object	

19.1.1.1.50. .spec.logStore.elasticsearch.proxy.resources

19.1.1.1.50.1. 설명

19.1.1.1.50.1.1. 유형

-

object

속성	유형	설명
limits	object	(선택 사항) 제한은 허용되는 최대 컴퓨팅 리소스 양을 설명합니다.
requests	object	(선택 사항) Requests는 필요한 최소 컴퓨팅 리소스 양을 설명합니다.

19.1.1.1.51. .spec.logStore.elasticsearch.proxy.resources.limits

19.1.1.1.51.1. 설명

19.1.1.1.51.1.1. 유형

-

object

19.1.1.1.52. .spec.logStore.elasticsearch.proxy.resources.requests

19.1.1.1.52.1. 설명

19.1.1.1.52.1.1. 유형

- **object**

19.1.1.1.53. .spec.logStore.elasticsearch.resources

19.1.1.1.53.1. 설명

19.1.1.1.53.1.1. 유형

- **object**

속성	유형	설명
limits	object	(선택 사항) 제한은 허용되는 최대 컴퓨팅 리소스 양을 설명합니다.
requests	object	(선택 사항) Requests는 필요한 최소 컴퓨팅 리소스 양을 설명합니다.

19.1.1.1.54. .spec.logStore.elasticsearch.resources.limits

19.1.1.1.54.1. 설명

19.1.1.1.54.1.1. 유형

- **object**

19.1.1.1.55. .spec.logStore.elasticsearch.resources.requests

19.1.1.1.55.1. 설명

19.1.1.1.55.1.1. 유형

- **object**

19.1.1.1.56. .spec.logStore.elasticsearch.storage

19.1.1.1.56.1. 설명**19.1.1.1.56.1.1. 유형**

●

object

속성	유형	설명
크기	object	프로비저닝 할 노드의 최대 스토리지 용량입니다.
storageClassName	string	(선택 사항) 노드의 PVC를 생성하는데 사용할 스토리지 클래스의 이름입니다.

19.1.1.1.57. .spec.logStore.elasticsearch.storage.size**19.1.1.1.57.1. 설명****19.1.1.1.57.1.1. 유형**

●

object

속성	유형	설명
형식	string	필요에 따라 형식 변경 Canonicalize에 대한 코멘트를 참조하십시오.
d	object	d는 d.Dec != nil인 inf.Dec 형식의 양입니다.
i	int	i는 int64 스케일링 형식의 양이며, d.Dec == nil
s	string	s는 재계산을 피하기 위해 이 수량의 생성된 값입니다.

19.1.1.1.58. .spec.logStore.elasticsearch.storage.size.d**19.1.1.1.58.1. 설명**

19.1.1.58.1.1. 유형

-

object

속성	유형	설명
Dec	object	

19.1.1.59. .spec.logStore.elasticsearch.storage.size.d.Dec

19.1.1.59.1. 설명

19.1.1.59.1.1. 유형

-

object

속성	유형	설명
scale	int	
Unscaled	object	

19.1.1.60. .spec.logStore.elasticsearch.storage.size.d.Dec.unscaled

19.1.1.60.1. 설명

19.1.1.60.1.1. 유형

-

object

속성	유형	설명
ABS	Word	서명
neg	bool	

19.1.1.61. .spec.logStore.elasticsearch.storage.size.d.Dec.unscaled.abs

19.1.1.61.1. 설명

19.1.1.61.1.1. 유형

•

Word**19.1.1.62. .spec.logStore.elasticsearch.storage.size.i****19.1.1.62.1. 설명****19.1.1.62.1.1. 유형**

•

int

속성	유형	설명
scale	int	
value	int	

19.1.1.63. .spec.logStore.elasticsearch.tolerations[]**19.1.1.63.1. 설명****19.1.1.63.1.1. 유형**

•

array

속성	유형	설명
effect	string	(선택 사항) Effect는 일치하는 테인트 효과를 나타냅니다. 빈은 모든 테인트 효과와 일치함을 의미합니다.
key	string	(선택 사항) Key는 허용 오차가 적용되는 테인트 키입니다. 빈은 모든 테인트 키와 일치함을 의미합니다.
operator	string	(선택 사항) Operator는 값과 키의 관계를 나타냅니다.

속성	유형	설명
tolerationSeconds	int	(선택 사항) TolerationSeconds는 허용 오차 (선택 사항)를 나타냅니다.
value	string	(선택 사항) 같은 톨러레이션이 일치하는 테인트 값입니다.

19.1.1.1.64. .spec.logStore.elasticsearch.tolerations[].tolerationSeconds

19.1.1.1.64.1. 설명

19.1.1.1.64.1.1. 유형

- **int**

19.1.1.1.65. .spec.logStore.lokistack

19.1.1.1.65.1. 설명

CloudEventStackStoreSpec은 로깅 스토리지로 **stack**을 사용하도록 **cluster-logging**을 설정하는데 사용됩니다. 동일한 네임 스페이스에서 기존**stack**을 가리킵니다.

19.1.1.1.65.1.1. 유형

- **object**

속성	유형	설명
name	string	ScanSettingStack 리소스의 이름입니다.

19.1.1.1.66. .spec.logStore.retentionPolicy

19.1.1.1.66.1. 설명

19.1.1.1.66.1.1. 유형

- **object**

속성	유형	설명
애플리케이션	object	
audit	object	
infra	object	

19.1.1.1.67. .spec.logStore.retentionPolicy.application

19.1.1.1.67.1. 설명

19.1.1.1.67.1.1. 유형

- **object**

속성	유형	설명
diskThresholdPercent	int	(선택 사항) ES 디스크 사용량의 임계값에 도달하면 이전 인덱스를 삭제해야 합니다(예: 75).
maxAge	string	(선택 사항)
namespaceSpec	array	(선택 사항) 지정된 최소 수명보다 오래된 문서를 삭제할 네임스페이스별 사양입니다.
pruneNamespacesInterval	string	(선택 사항) 새 정리 네임스페이스 작업을 실행하는 빈도

19.1.1.1.68. .spec.logStore.retentionPolicy.application.namespaceSpec[]

19.1.1.1.68.1. 설명

19.1.1.1.68.1.1. 유형

- **array**

속성	유형	설명

속성	유형	설명
minAge	string	(선택 사항) 이 MinAge (예: 1d)보다 오래된 네임스페이스와 일치하는 레코드를 삭제합니다.
namespace	string	MinAge보다 오래된 로그를 삭제하는 대상 네임스페이스 (기본값: 7d)

19.1.1.1.69. .spec.logStore.retentionPolicy.audit

19.1.1.1.69.1. 설명

19.1.1.1.69.1.1. 유형

- **object**

속성	유형	설명
diskThresholdPercent	int	(선택 사항) ES 디스크 사용량의 임계값에 도달하면 이전 인덱스를 삭제해야 합니다(예: 75).
maxAge	string	(선택 사항)
namespaceSpec	array	(선택 사항) 지정된 최소 수명보다 오래된 문서를 삭제할 네임스페이스별 사양입니다.
pruneNamespacesInterval	string	(선택 사항) 새 정리 네임스페이스 작업을 실행하는 빈도

19.1.1.1.70. .spec.logStore.retentionPolicy.audit.namespaceSpec[]

19.1.1.1.70.1. 설명

19.1.1.1.70.1.1. 유형

- **array**

속성	유형	설명

속성	유형	설명
minAge	string	(선택 사항) 이 MinAge (예: 1d)보다 오래된 네임스페이스와 일치하는 레코드를 삭제합니다.
namespace	string	MinAge보다 오래된 로그를 삭제하는 대상 네임스페이스 (기본값: 7d)

19.1.1.1.71. .spec.logStore.retentionPolicy.infra

19.1.1.1.71.1. 설명

19.1.1.1.71.1.1. 유형

- **object**

속성	유형	설명
diskThresholdPercent	int	(선택 사항) ES 디스크 사용량의 임계값에 도달하면 이전 인덱스를 삭제해야 합니다(예: 75).
maxAge	string	(선택 사항)
namespaceSpec	array	(선택 사항) 지정된 최소 수명보다 오래된 문서를 삭제할 네임스페이스별 사양입니다.
pruneNamespacesInterval	string	(선택 사항) 새 정리 네임스페이스 작업을 실행하는 빈도

19.1.1.1.72. .spec.logStore.retentionPolicy.infra.namespaceSpec[]

19.1.1.1.72.1. 설명

19.1.1.1.72.1.1. 유형

- **array**

속성	유형	설명

속성	유형	설명
minAge	string	(선택 사항) 이 MinAge (예: 1d)보다 오래된 네임스페이스와 일치하는 레코드를 삭제합니다.
namespace	string	MinAge보다 오래된 로그를 삭제하는 대상 네임스페이스 (기본값: 7d)

19.1.1.1.73. .spec.visualization

19.1.1.1.73.1. 설명

이는 **Log visualization(Kibana)**에 대한 정보를 포함하는 구조입니다.

19.1.1.1.73.1.1. 유형

- **object**

속성	유형	설명
kibana	object	Kibana 시각화 구성 요소 사양
type	string	구성할 시각화 유형입니다.

19.1.1.1.74. .spec.visualization.kibana

19.1.1.1.74.1. 설명

19.1.1.1.74.1.1. 유형

- **object**

속성	유형	설명
nodeSelector	object	Pod가 예약된 노드를 정의합니다.
proxy	object	Kibana 프록시 구성 요소 사양
replicas	int	Kibana 배포에 배포할 인스턴스 수

속성	유형	설명
resources	object	(선택 사항) Kibana의 리소스 요구 사항입니다.
허용 오차	array	

19.1.1.1.75. .spec.visualization.kibana.nodeSelector

19.1.1.1.75.1. 설명

19.1.1.1.75.1.1. 유형

- **object**

19.1.1.1.76. .spec.visualization.kibana.proxy

19.1.1.1.76.1. 설명

19.1.1.1.76.1.1. 유형

- **object**

속성	유형	설명
resources	object	

19.1.1.1.77. .spec.visualization.kibana.proxy.resources

19.1.1.1.77.1. 설명

19.1.1.1.77.1.1. 유형

- **object**

속성	유형	설명
limits	object	(선택 사항) 제한은 허용되는 최대 컴퓨팅 리소스 양을 설명합니다.

속성	유형	설명
requests	object	(선택 사항) Requests는 필요한 최소 컴퓨팅 리소스 양을 설명합니다.

19.1.1.1.78. .spec.visualization.kibana.proxy.resources.limits

19.1.1.1.78.1. 설명

19.1.1.1.78.1.1. 유형

- object

19.1.1.1.79. .spec.visualization.kibana.proxy.resources.requests

19.1.1.1.79.1. 설명

19.1.1.1.79.1.1. 유형

- object

19.1.1.1.80. .spec.visualization.kibana.replicas

19.1.1.1.80.1. 설명

19.1.1.1.80.1.1. 유형

- int

19.1.1.1.81. .spec.visualization.kibana.resources

19.1.1.1.81.1. 설명

19.1.1.1.81.1.1. 유형

- object

속성	유형	설명
limits	object	(선택 사항) 제한은 허용되는 최대 컴퓨팅 리소스 양을 설명합니다.
requests	object	(선택 사항) Requests는 필요한 최소 컴퓨팅 리소스 양을 설명합니다.

19.1.1.1.82. .spec.visualization.kibana.resources.limits

19.1.1.1.82.1. 설명

19.1.1.1.82.1.1. 유형

- **object**

19.1.1.1.83. .spec.visualization.kibana.resources.requests

19.1.1.1.83.1. 설명

19.1.1.1.83.1.1. 유형

- **object**

19.1.1.1.84. .spec.visualization.kibana.tolerations[]

19.1.1.1.84.1. 설명

19.1.1.1.84.1.1. 유형

- **array**

속성	유형	설명
effect	string	(선택 사항) Effect는 일치하는 테인트 효과를 나타냅니다. 빈은 모든 테인트 효과와 일치함을 의미합니다.
key	string	(선택 사항) Key는 허용 오차가 적용되는 테인트 키입니다. 빈은 모든 테인트 키와 일치함을 의미합니다.

속성	유형	설명
operator	string	(선택 사항) Operator는 값과 키의 관계를 나타냅니다.
tolerationSeconds	int	(선택 사항) TolerationSeconds는 허용 오차(선택 사항)를 나타냅니다.
value	string	(선택 사항) 같은 톨러레이션이 일치하는 테인트 값을 합니다.

19.1.1.1.85. .spec.visualization.kibana.tolerations[].tolerationSeconds

19.1.1.1.85.1. 설명

19.1.1.1.85.1.1. 유형

- **int**

19.1.1.1.86. .status

19.1.1.1.86.1. 설명

ClusterLoggingStatus는 ClusterLogging의 관찰 상태를 정의합니다.

19.1.1.1.86.1.1. 유형

- **object**

속성	유형	설명
컬렉션	object	(선택 사항)
conditions	object	(선택 사항)
큐레이션	object	(선택 사항)
logStore	object	(선택 사항)
시각화	object	(선택 사항)

19.1.1.1.87. .status.collection**19.1.1.1.87.1. 설명****19.1.1.1.87.1.1. 유형**

•

object

속성	유형	설명
logs	object	(선택 사항)

19.1.1.1.88. .status.collection.logs**19.1.1.1.88.1. 설명****19.1.1.1.88.1.1. 유형**

•

object

속성	유형	설명
fluentdStatus	object	(선택 사항)

19.1.1.1.89. .status.collection.logs.fluentdStatus**19.1.1.1.89.1. 설명****19.1.1.1.89.1.1. 유형**

•

object

속성	유형	설명
clusterCondition	object	(선택 사항)
daemonSet	string	(선택 사항)
노드	object	(선택 사항)

속성	유형	설명
pods	string	(선택 사항)

19.1.1.1.90. .status.collection.logs.fluentdStatus.clusterCondition

19.1.1.1.90.1. 설명

Operator-sdk generate crds 에서 map-of-slice를 허용하지 않으므로 이름이 지정된 유형을 사용해야 합니다.

19.1.1.1.90.1.1. 유형

- object

19.1.1.1.91. .status.collection.logs.fluentdStatus.nodes

19.1.1.1.91.1. 설명

19.1.1.1.91.1.1. 유형

- object

19.1.1.1.92. .status.conditions

19.1.1.1.92.1. 설명

19.1.1.1.92.1.1. 유형

- object

19.1.1.1.93. .status.curation

19.1.1.1.93.1. 설명

19.1.1.1.93.1.1. 유형

- object

속성	유형	설명
curatorStatus	array	(선택 사항)

19.1.1.1.94. .status.curation.curatorStatus[]**19.1.1.1.94.1. 설명****19.1.1.1.94.1.1. 유형**

- **array**

속성	유형	설명
clusterCondition	object	(선택 사항)
cronjobs	string	(선택 사항)
스케줄	string	(선택 사항)
일시 중단됨	bool	(선택 사항)

19.1.1.1.95. .status.curation.curatorStatus[].clusterCondition**19.1.1.1.95.1. 설명**

Operator-sdk generate crds 에서 map-of-slice를 허용하지 않으므로 이름이 지정된 유형을 사용해야 합니다.

19.1.1.1.95.1.1. 유형

- **object**

19.1.1.1.96. .status.logStore**19.1.1.1.96.1. 설명****19.1.1.1.96.1.1. 유형**

-

object

속성	유형	설명
elasticsearchStatus	array	(선택 사항)

19.1.1.1.97. .status.logStore.elasticsearchStatus[]

19.1.1.1.97.1. 설명

19.1.1.1.97.1.1. 유형

-

array

속성	유형	설명
cluster	object	(선택 사항)
clusterConditions	object	(선택 사항)
clusterHealth	string	(선택 사항)
clusterName	string	(선택 사항)
Deployments	array	(선택 사항)
nodeConditions	object	(선택 사항)
nodeCount	int	(선택 사항)
pods	object	(선택 사항)
replicaSets	array	(선택 사항)
shardAllocationEnabled	string	(선택 사항)
statefulSets	array	(선택 사항)

19.1.1.1.98. .status.logStore.elasticsearchStatus[].cluster

19.1.1.1.98.1. 설명

19.1.1.98.1.1. 유형

●

object

속성	유형	설명
activePrimaryShards	int	Elasticsearch 클러스터의 활성 기본 Shard 수
activeShards	int	Elasticsearch 클러스터의 활성 Shard 수
initializingShards	int	Elasticsearch 클러스터의 Initializing Shards 수
numDataNodes	int	Elasticsearch 클러스터의 데이터 노드 수
numNodes	int	Elasticsearch 클러스터의 노드 수
pendingTasks	int	
relocatingShards	int	Elasticsearch 클러스터의 재배치 Shard 수
status	string	Elasticsearch 클러스터의 현재 상태
unassignedShards	int	Elasticsearch 클러스터의 할당되지 않은 Shard 수

19.1.1.99. .status.logStore.elasticsearchStatus[].clusterConditions**19.1.1.99.1. 설명**

●

object**19.1.1.100. .status.logStore.elasticsearchStatus[].deployments[]****19.1.1.100.1. 설명****19.1.1.100.1.1. 유형**

- array

19.1.1.1.101. .status.logStore.elasticsearchStatus[].nodeConditions

19.1.1.1.101.1. 설명

19.1.1.1.101.1.1. 유형

- object

19.1.1.1.102. .status.logStore.elasticsearchStatus[].pods

19.1.1.1.102.1. 설명

19.1.1.1.102.1.1. 유형

- object

19.1.1.1.103. .status.logStore.elasticsearchStatus[].replicaSets[]

19.1.1.1.103.1. 설명

19.1.1.1.103.1.1. 유형

- array

19.1.1.1.104. .status.logStore.elasticsearchStatus[].statefulSets[]

19.1.1.1.104.1. 설명

19.1.1.1.104.1.1. 유형

- array

19.1.1.1.105. .status.visualization

19.1.1.1.105.1. 설명

19.1.1.1.105.1.1. 유형

-

object

속성	유형	설명
kibanaStatus	array	(선택 사항)

19.1.1.1.106. .status.visualization.kibanaStatus[]**19.1.1.1.106.1. 설명****19.1.1.1.106.1.1. 유형**

-

array

속성	유형	설명
clusterCondition	object	(선택 사항)
Deployment	string	(선택 사항)
pods	string	(선택 사항) 시각화 구성 요소에 대한 각 Kibana Pod의 상태
replicaSets	array	(선택 사항)
replicas	int	(선택 사항)

19.1.1.1.107. .status.visualization.kibanaStatus[].clusterCondition**19.1.1.1.107.1. 설명****19.1.1.1.107.1.1. 유형**

-

object**19.1.1.1.108. .status.visualization.kibanaStatus[].replicaSets[]****19.1.1.1.108.1. 설명**

19.1.1.108.1.1. 유형

- **array**

20장. 용어집

이 용어집은 로깅 설명서에 사용되는 일반적인 용어를 정의합니다.

주석

주석을 사용하여 메타데이터를 오브젝트에 연결할 수 있습니다.

Red Hat OpenShift Logging Operator

Red Hat OpenShift Logging Operator는 애플리케이션, 인프라 및 감사 로그의 수집 및 전달을 제어하는 API 세트를 제공합니다.

CR(사용자 정의 리소스)

CR은 Kubernetes API의 확장입니다. 로깅 및 로그 전달을 구성하려면 **ClusterLogging** 및 **ClusterLogForwarder** 사용자 정의 리소스를 사용자 지정할 수 있습니다.

이벤트 라우터

이벤트 라우터는 **OpenShift Container Platform** 이벤트를 감시하는 Pod입니다. 로깅을 사용하여 로그를 수집합니다.

fluentd

Fluentd는 각 **OpenShift Container Platform** 노드에 상주하는 로그 수집기입니다. 애플리케이션, 인프라 및 감사 로그를 수집하여 다른 출력으로 전달합니다.

가비지 컬렉션

가비지 컬렉션은 실행 중인 Pod에서 참조하지 않는 종료 컨테이너 및 이미지와 같은 클러스터 리소스를 정리하는 프로세스입니다.

Elasticsearch

Elasticsearch는 분산 검색 및 분석 엔진입니다. **OpenShift Container Platform**은 **Elasticsearch**를 로깅의 기본 로그 저장소로 사용합니다.

OpenShift Elasticsearch Operator

OpenShift Elasticsearch Operator은 **OpenShift Container Platform**에서 **Elasticsearch** 클러스터를 실행하는 데 사용됩니다. **OpenShift Elasticsearch Operator**는 **Elasticsearch** 클러스터 작업에 대한 셀프 서비스를 제공하며 로깅에서 사용합니다.

인덱싱

인덱싱은 데이터를 신속하게 찾고 액세스하는 데 사용되는 데이터 구조 기술입니다. 인덱싱은 쿼리를 처리할 때 필요한 디스크 액세스 양을 최소화하여 성능을 최적화합니다.

JSON 로깅

로그 전달 API를 사용하면 JSON 로그를 구조화된 오브젝트로 구문 분석하고 로그 전달 API에서 지원하는 로깅 관리 Elasticsearch 또는 기타 타사 시스템으로 전달할 수 있습니다.

Kibana

Kibana는 히스토그램, 선 그래프 및 원형 차트를 통해 Elasticsearch 데이터를 쿼리, 검색 및 시각화하는 브라우저 기반 콘솔 인터페이스입니다.

Kubernetes API 서버

Kubernetes API 서버는 API 오브젝트의 데이터를 검증하고 구성합니다.

라벨

레이블은 Pod와 같은 오브젝트 서브 세트를 구성하고 선택하는 데 사용할 수 있는 키-값 쌍입니다.

로깅

로깅을 사용하면 클러스터 전체에서 애플리케이션, 인프라 및 감사 로그를 집계할 수 있습니다. 또한 기본 로그 저장소로 저장하고 타사 시스템으로 전달한 다음 기본 로그 저장소에서 저장된 로그를 쿼리 및 시각화할 수 있습니다.

로깅 수집기

로깅 수집기는 클러스터에서 로그를 수집하여 포맷한 후 로그 저장소 또는 타사 시스템으로 전달합니다.

로그 저장소

로그 저장소는 집계된 로그를 저장하는 데 사용됩니다. 내부 로그 저장소를 사용하거나 로그를 외부 로그 저장소로 전달할 수 있습니다.

로그 시각화 프로그램

로그 시각화 프로그램은 로그, 그래프, 차트 및 기타 지표와 같은 정보를 보는 데 사용할 수 있는 UI(사용자 인터페이스) 구성 요소입니다.

노드

노드는 OpenShift Container Platform 클러스터의 작업자 시스템입니다. 노드는 VM(가상 머신)

또는 물리적 머신입니다.

Operator

Operator은 OpenShift Container Platform 클러스터에서 Kubernetes 애플리케이션을 패키징, 배포 및 관리하는 기본 방법입니다. **Operator**은 사람의 운영 지식을 패키지하고 고객과 공유하는 소프트웨어로 인코딩합니다.

Pod

Pod은 Kubernetes에서 가장 작은 논리 단위입니다. **Pod**는 하나 이상의 컨테이너로 구성되며 작업자 노드에서 실행됩니다.

RBAC(역할 기반 액세스 제어)

RBAC은 클러스터 사용자 및 워크로드가 역할을 실행하는 데 필요한 리소스에만 액세스할 수 있도록 하는 핵심 보안 제어입니다.

shard

Elasticsearch은 **Fluentd**의 로그 데이터를 데이터 저장소 또는 인덱스로 구성한 다음 각 인덱스를 **shard**라는 여러 조각으로 세분화합니다.

taint

테인트를 사용하면 **Pod**가 적절한 노드에 예약됩니다. 노드에 하나 이상의 테인트를 적용할 수 있습니다.

톨러레이션

Pod에 허용 오차를 적용할 수 있습니다. 허용 오차를 사용하면 스케줄러에서 일치하는 테인트를 사용하여 **Pod**를 예약할 수 있습니다.

웹 콘솔

OpenShift Container Platform을 관리할 UI(사용자 인터페이스)입니다.