OpenShift Container Platform 4.8

로깅

OpenShift Logging 설치, 사용법, 릴리스 정보

Last Updated: 2023-06-12

OpenShift Container Platform 4.8 로깅

OpenShift Logging 설치, 사용법, 릴리스 정보

법적 공지

초록

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

차례

1장. 로깅의 릴리스 노트

로깅 호환성

Red Hat OpenShift의 로깅 하위 시스템은 핵심 OpenShift Container Platform과는 별도의 릴리스 사이클과 함께 설치 가능한 구성 요소로 제공됩니다. Red Hat OpenShift Container Platform 라이프 사이클 정책은 릴리스 호환성에 대해 간략하게 설명합니다.

1.1. LOGGING 5.4.9

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.4.9 가 포함되어 있습니다.

1.1.1. 버그 수정

- 이번 업데이트 이전에는 Fluentd 수집기에서 사용되지 않은 구성 매개변수에 대해 경고를 표시했습니다. 이번 업데이트에서는 이러한 구성 매개변수와 경고 메시지를 제거합니다. (로그-3074)
- 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-3306)

1.1.2. 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

1.2. LOGGING 5.4.8

이번 릴리스에는 RHSA-2022:7435-OpenShift Logging 버그 수정 릴리스 5.4.8 이 포함되어 있습니다.

1.2.1. 버그 수정

없음.

1.2.2. CVE

- CVE-2016-3709
- CVE-2020-35525
- CVE-2020-35527
- CVE-2020-36518
- CVE-2022-1304
- CVE-2022-2509
- CVE-2022-3515
- CVE-2022-22624
- CVE-2022-22628
- CVE-2022-22629
- CVE-2022-22662
- CVE-2022-26700
- CVE-2022-26709
- CVE-2022-26710
- CVE-2022-26716
- CVE-2022-26717
- CVE-2022-26719
- CVE-2022-30293
- CVE-2022-32149
- CVE-2022-37434
- CVE-2022-40674
- CVE-2022-42003
- CVE-2022-42004

1.3. LOGGING 5.4.6

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.4.6 이 포함되어 있습니다.

1.3.1. 버그 수정

- 이번 업데이트 이전에는 Fluentd가 Kubernetes 플랫폼이 로그 파일을 순환하여 더 이상 로그 메시지를 읽지 않는 경우가 있었습니다. 이번 업데이트에서는 업스트림 개발 팀이 제안한 구성 매개변수를 설정하여 수정됩니다. (로그-2792)

- 이번 업데이트 이전에는 **ClusterLogForwarder** 사용자 정의 리소스에 JSON 구문 분석이 정의된 경우 각 롤오버 작업에 빈 인덱스가 생성되었습니다. 이번 업데이트를 통해 새 인덱스가 비어 있지 않습니다. (LOG-2823)

- 이번 업데이트 이전에는 Kibana 사용자 정의 리소스를 삭제한 경우 OpenShift Container Platform 웹 콘솔에 Kibana에 대한 링크가 계속 표시되었습니다. 이번 업데이트를 통해 Kibana 사용자 정의 리소스를 제거하면 해당 링크도 제거됩니다. (로그-3054)

1.3.2. 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. 로깅 5.4.5

이번 릴리스에는 RHSA-2022:6183-OpenShift Logging 버그 수정 릴리스 5.4.5 가 포함되어 있습니다.

1.4.1. 버그 수정

- 이번 업데이트 이전에는 Operator에서 Pod가 준비되었는지 확인하지 않아 클러스터를 다시 시작하는 동안 클러스터가 작동하지 않는 상태에 도달했습니다. 이번 업데이트를 통해 Operator는 재시작 중에 새 Pod를 계속 진행하기 전에 새 Pod를 준비 상태로 표시하여 문제를 해결합니다. (LOG-2881)

- 이번 업데이트 이전에는 여러 줄 오류 감지 기능을 추가하여 내부 라우팅이 변경되어 레코드를 잘못된 대상으로 전달했습니다. 이번 업데이트를 통해 내부 라우팅이 잘못되었습니다. (LOG-2946)

- 이번 업데이트 이전에는 Operator에서 인용된 부울 값을 사용하여 인덱스 설정 JSON 응답을 디코딩할 수 없어 오류가 발생했습니다. 이번 업데이트를 통해 Operator는 이 JSON 응답을 적절하게 디코딩할 수 있습니다. (LOG-3009)

- 이번 업데이트 이전에는 Elasticsearch 인덱스 템플릿이 잘못된 유형의 라벨 필드를 정의했습니다. 이번 변경으로 인해 로그 수집기가 전달하는 예상 유형과 일치하도록 해당 템플릿이 업데이트됩니다. (LOG-2972)

1.4.2. CVE

- CVE-2022-1292
- CVE-2022-1586
- CVE-2022-1785
- CVE-2022-1897
- CVE-2022-1927
- CVE-2022-2068
- CVE-2022-2097
- CVE-2022-30631

1.5. 로깅 5.4.4

이 릴리스에는 RHBA-2022:5907-OpenShift Logging 버그 수정 릴리스 5.4.4 가 포함되어 있습니다.

1.5.1. 버그 수정

- 이번 업데이트 이전에는 Elasticsearch에 올바르지 않은 문자가 잘못 표시되었습니다. 이번 업데이트를 통해 Elasticsearch는 모든 유효한 UTF-8 기호를 올바르게 표시합니다. (LOG-2794)
- 이번 업데이트 이전에는 Fluentd에 올바르지 않은 문자가 잘못 표시되었습니다. 이번 업데이트를 통해 Fluentd는 모든 유효한 UTF-8 기호를 올바르게 표시합니다. (LOG-2657)
- 이번 업데이트 이전에는 수집기의 지표 서버가 환경 값에서 노출하는 값을 사용하여 주소에 바인딩하려고 했습니다. 이 변경 사항은 사용 가능한 모든 인터페이스에 바인딩할 구성을 수정합니다. (LOG-2821)
- 이번 업데이트 이전에는 **cluster-logging** Operator를 클러스터에 의존하여 보안을 생성했습니다. OpenShift Container Platform 4.11에서 이 클러스터 동작이 변경되어 로깅 배포가 실패했습니다. 이번 업데이트를 통해 필요한 경우 **cluster-logging** Operator가 보안을 생성하여 문제를 해결합니다. (LOG-2840)

1.5.2. CVE

- CVE-2022-21540
- CVE-2022-21541
- CVE-2022-34169

1.6. 로깅 5.4.3

이 릴리스에는 RHBA-2022:5556-OpenShift Logging 버그 수정 릴리스 5.4.3 이 포함되어 있습니다.

1.6.1. Elasticsearch Operator 사용 중단 알림

로깅 하위 시스템 5.4.3에서 Elasticsearch Operator는 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. Elasticsearch Operator를 사용하여 기본 로그 스토리지를 관리하는 대신 Loki Operator를 사용할 수 있습니다.

1.6.2. 버그 수정

- 이번 업데이트 이전에는 OpenShift Logging 대시보드에 모든 활성 shard 대신 활성 기본 shard 수가 표시되었습니다. 이번 업데이트를 통해 대시보드에 활성 shard가 모두 표시됩니다. (LOG-2781)
- 이번 업데이트 이전에는 **elasticsearch-operator** 에서 사용하는 라이브러리의 버그에 서비스 거부 취약점이 포함되었습니다. 이번 업데이트를 통해 라이브러리가 이 취약점을 포함하지 않는 버전으로 업데이트되었습니다. (LOG-2816)
- 이번 업데이트 이전에는 로그를 로키로 전달하도록 Vector를 구성할 때 사용자 정의 전달 토큰을 설정하거나 Loki가 TLS가 활성화된 경우 기본 토큰을 사용할 수 없었습니다. 이번 업데이트를 통해 Vector는 TLS가 활성화된 토큰을 사용하여 로키에 로그를 전달할 수 있습니다. (LOG-2786
- 이번 업데이트 이전에는 **oauth-proxy** 이미지를 선택할 때 ElasticSearch Operator가 **ImageStream** 사용자 정의 리소스의 **referencePolicy** 속성을 생략했습니다. 이로 인해 특정 환경에서 Kibana 배포가 실패했습니다. 이번 업데이트를 통해 **referencePolicy** 를 사용하면 문제가 해결되고 Operator에서 Kibana를 성공적으로 배포할 수 있습니다. (LOG-2791)
- 이번 업데이트 이전에는 **ClusterLogForwarder** 사용자 정의 리소스에 대한 경고 규칙에 여러 개의 전달 출력이 고려되지 않았습니다. 이번 업데이트에서는 이러한 문제가 해결되었습니다. (LOG-2640)
- 이번 업데이트 이전에는 로그를 Amazon ONTAP에 전달하도록 구성된 클러스터는 거부된 로그 파일을 임시 스토리지로 전달하여 시간이 지남에 따라 클러스터 불안정을 초래했습니다. 이번 업데이트를 통해 Gradle의 청크 백업이 비활성화되어 문제를 해결합니다. (LOG-2768)

1.6.3. CVE

예 1.1. CVE를 확장하려면 클릭합니다.

- CVE-2020-28915
- CVE-2021-40528
- CVE-2022-1271
- CVE-2022-1621
- CVE-2022-1629
- CVE-2022-22576
- CVE-2022-25313
- CVE-2022-25314
- CVE-2022-26691
- CVE-2022-27666
- CVE-2022-27774
- CVE-2022-27776
- CVE-2022-27782

- CVE-2022-29824

1.7. 로깅 5.4.2

이 릴리스에는 RHBA-2022:4874-OpenShift Logging 버그 수정 릴리스 5.4.2 가 포함되어 있습니다.

1.7.1. 버그 수정

- 이번 업데이트 이전에는 **oc edit** 를 사용하여 수집기 구성을 편집하기 때문에 공백을 일관하지 않게 사용했습니다. 이러한 변경으로 인해 Operator의 업데이트 전에 구성을 표준화하고 포맷하여 **oc edit** 를 사용하여 쉽게 편집할 수 있습니다. (LOG-2319)

- 이번 업데이트 이전에는 **FluentdNodeDown** 경고가 message 섹션에 인스턴스 라벨을 적절하게 제공할 수 없었습니다. 이번 업데이트에서는 부분적인 인스턴스 오류가 발생하는 경우 인스턴스 레이블을 제공하기 위해 경고 규칙을 수정하여 문제를 해결합니다. (LOG-2607)

- 이번 업데이트 이전에는 제품의 지원대로 문서화된'과 같은 여러 로그 수준이 없었습니다. 이번 업데이트에서는 불일치가 수정되어 문서화된 로그 수준이 제품에서 지원됩니다. (LOG-2033)

1.7.2. CVE

예 1.2. CVE를 확장하려면 클릭합니다.

- CVE-2018-25032
- CVE-2020-0404
- CVE-2020-4788
- CVE-2020-13974
- CVE-2020-19131
- CVE-2020-27820
- CVE-2021-0941
- CVE-2021-3612
- CVE-2021-3634
- CVE-2021-3669
- CVE-2021-3737
- CVE-2021-3743
- CVE-2021-3744
- CVE-2021-3752
- CVE-2021-3759
- CVE-2021-3764

- CVE-2021-3772
- CVE-2021-3773
- CVE-2021-4002
- CVE-2021-4037
- CVE-2021-4083
- CVE-2021-4157
- CVE-2021-4189
- CVE-2021-4197
- CVE-2021-4203
- CVE-2021-20322
- CVE-2021-21781
- CVE-2021-23222
- CVE-2021-26401
- CVE-2021-29154
- CVE-2021-37159
- CVE-2021-41617
- CVE-2021-41864
- CVE-2021-42739
- CVE-2021-43056
- CVE-2021-43389
- CVE-2021-43976
- CVE-2021-44733
- CVE-2021-45485
- CVE-2021-45486
- CVE-2022-0001
- CVE-2022-0002
- CVE-2022-0286
- CVE-2022-0322
- CVE-2022-1011

- CVE-2022-1271

1.8. 로깅 5.4.1

이번 릴리스에는 RHSA-2022:216-OpenShift Logging 버그 수정 릴리스 5.4.1 가 포함되어 있습니다.

1.8.1. 버그 수정

- 이번 업데이트 이전에는 내보내기자가 실행되는 동안 생성된 로그만 로그 파일 지표 내보내기에서 보고했기 때문에 로그 증가 데이터가 부정확했습니다. 이번 업데이트에서는 **/var/log/pods** 를 모니터링하여 이 문제를 해결합니다. (LOG-2442)

- 이번 업데이트 이전에는 fluentd forward receiver에 로그를 전달할 때 오래된 연결을 계속 사용하려고 하기 때문에 수집기가 차단됩니다. 이번 릴리스에서는 수집기에서 연결을 재활용하고 실패한 메시지를 합리적인 시간 내에 보내도록 **keepalive_timeout** 값이**30초(30**초)로 설정되어 있습니다. (LOG-2534)

- 이번 업데이트 이전에는 Kubernetes 네임스페이스를 사용하여 로그에 대한 제한된 액세스 권한을 읽기 위해 게이트웨이 구성 요소의 테넌시가 있어 "감사" 및 일부 "인프라" 로그를 읽을 수 없습니다. 이번 업데이트를 통해 프록시는 관리자 액세스 권한이 있는 사용자를 올바르게 감지하고 네임스페이스 없이 로그에 액세스할 수 있습니다. (LOG-2448)

- 이번 업데이트 이전에는 **system:serviceaccount:openshift-monitoring:prometheus-k8s** 서비스 계정에 clusterrole 및 **cluster role binding 의 클러스터** 수준 권한이 있었습니다. 이번 업데이트에서는 서비스 계정의 이름이 role 및 rolebinding을 사용하여 **openshift-logging** 네임스페이스로 제한됩니다. (LOG-2437)

- 이번 업데이트 이전에는 Linux 감사 로그 시간 구문 분석이 키/값 쌍의 서수 위치에 의존했습니다. 이번 업데이트에서는 정규식을 사용하여 시간 항목을 찾도록 구문 분석이 변경되었습니다. (LOG-2321)

1.8.2. CVE

예 1.3. CVE를 확장하려면 클릭합니다.

- CVE-2018-25032
- CVE-2021-4028
- CVE-2021-37136
- CVE-2021-37137
- CVE-2021-43797
- CVE-2022-0778
- CVE-2022-1154
- CVE-2022-1271
- CVE-2022-21426
- CVE-2022-21434

- CVE-2022-21443
- CVE-2022-21476
- CVE-2022-21496
- CVE-2022-21698
- CVE-2022-25636

1.9. LOGGING 5.4

다음 권고는 5.4 로깅에 사용할 수 있습니다. Red Hat OpenShift 릴리스 5.4의 로깅 하위 시스템

1.9.1. 기술 프리뷰

중요

벡터는 기술 프리뷰 기능 전용입니다. 기술 프리뷰 기능은 Red Hat 프로덕션 서비스 수준 계약(SLA)에서 지원되지 않으며 기능적으로 완전하지 않을 수 있습니다. 따라서 프로덕션 환경에서 사용하는 것은 권장하지 않습니다. 이러한 기능을 사용하면 향후 제품 기능을 조기에 이용할 수 있어 개발 과정에서 고객이 기능을 테스트하고 피드백을 제공할 수 있습니다.

Red Hat 기술 프리뷰 기능의 지원 범위에 대한 자세한 내용은 https://access.redhat.com/support/offerings/techpreview/를 참조하십시오.

1.9.2. 벡터 정보

vector는 로깅 하위 시스템의 현재 기본 수집기에 대한 기술 프리뷰 대안으로 제공되는 로그 수집기입니다.

다음 출력이 지원됩니다.

- **elasticsearch**. 외부 Elasticsearch 인스턴스입니다. **elasticsearch** 출력은 TLS 연결을 사용할 수 있습니다.
- **kafka**. Kafka 브로커. **kafka** 출력은 비보안 또는 TLS 연결을 사용할 수 있습니다.
- **loki**. 수평으로 확장 가능한 고가용성 다중 테넌트 로그 집계 시스템인 Loki입니다.

1.9.2.1. 벡터 활성화

벡터는 기본적으로 활성화되어 있지 않습니다. 다음 단계를 사용하여 OpenShift Container Platform 클러스터에서 벡터를 활성화합니다.

중요

벡터는 FIPS 활성화된 클러스터를 지원하지 않습니다.

사전 요구 사항

- OpenShift Container Platform: 4.10

- Red Hat OpenShift의 로깅 하위 시스템: 5.4
- FIPS 비활성화

프로세스

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

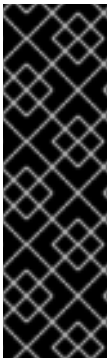
```
$ oc -n openshift-logging edit ClusterLogging instance
```

2. **ClusterLogging** 사용자 정의 리소스(CR)에 **logging.openshift.io/preview-vector-collector: enabled** 주석을 추가합니다.
3. **ClusterLogging** 사용자 정의 리소스(CR)에 컬렉션 유형으로 **벡터** 를 추가합니다.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: "openshift-logging"
  annotations:
    logging.openshift.io/preview-vector-collector: enabled
spec:
  collection:
  logs:
    type: "vector"
    vector: {}
```

추가 리소스

- vector 문서



중요

Loki Operator는 기술 프리뷰 기능 전용입니다. 기술 프리뷰 기능은 Red Hat 프로덕션 서비스 수준 계약(SLA)에서 지원되지 않으며 기능적으로 완전하지 않을 수 있습니다. 따라서 프로덕션 환경에서 사용하는 것은 권장하지 않습니다. 이러한 기능을 사용하면 향후 제품 기능을 조기에 이용할 수 있어 개발 과정에서 고객이 기능을 테스트하고 피드백을 제공할 수 있습니다.

Red Hat 기술 프리뷰 기능의 지원 범위에 대한 자세한 내용은 https://access.redhat.com/support/offerings/techpreview/를 참조하십시오.

1.9.3. Loki 정보

Loki는 수평적으로 확장 가능하고 가용성이 높은 멀티 테넌트 로그 집계 시스템으로 현재 로깅 하위 시스템의 로그 저장소로 Elasticsearch에 대한 대안으로 제공됩니다.

추가 리소스

- Loki 문서

1.9.3.1. Lokistack 배포

OpenShift Container Platform 웹 콘솔을 사용하여 LokiOperator를 설치할 수 있습니다.

사전 요구 사항

- OpenShift Container Platform: 4.10
- Red Hat OpenShift의 로깅 하위 시스템: 5.4

OpenShift Container Platform 웹 콘솔을 사용하여 LokiOperator를 설치하려면 다음을 수행합니다.

1. LokiOperator를 설치합니다.
   a. OpenShift Container Platform 웹 콘솔에서 **Operator → OperatorHub**를 클릭합니다.
   b. 사용 가능한 Operator 목록에서 **LokiOperator** 를 선택하고 **설치**를 클릭합니다.
   c. **설치 모드**에서 **클러스터의 모든 네임스페이스**를 선택합니다.
   d. **설치된 네임스페이스** 에서 **openshift-operators-redhat** 을 선택합니다. **openshift-operators-redhat** 네임스페이스를 지정해야 합니다. **openshift-operators** 네임스페이스에 신뢰할 수 없는 Community Operator가 포함될 수 있고, 여기에서 OpenShift Container Platform 지표와 동일한 이름의 지표를 게시하면 충돌이 발생합니다.
   e. **이 네임스페이스에서 Operator 권장 클러스터 모니터링 사용**을 선택합니다. 이 옵션은 네임스페이스 오브젝트에서 **openshift.io/cluster-monitoring: "true"** 레이블을 설정합니다. 클러스터 모니터링이 **openshift-operators-redhat** 네임스페이스를 스크랩하도록 하려면 이 옵션을 선택해야 합니다.
   f. **승인 전략**을 선택합니다.
      - **자동** 전략을 사용하면 Operator 새 버전이 준비될 때 OLM(Operator Lifecycle Manager)이 자동으로 Operator를 업데이트할 수 있습니다.
      - **수동** 전략을 사용하려면 적절한 자격 증명을 가진 사용자가 Operator 업데이트를 승인해야 합니다.
   g. **설치**를 클릭합니다.
   h. LokiOperator가 설치되었는지 확인합니다. **Operator → 설치된 Operator 페이지를 방문하여** "LokiOperator"를 찾습니다.
   i. 상태가 성공인 모든 프로젝트에 **LokiOperator** 가 나열되어 있는지 **확인합니다**.

1.9.4. 버그 수정

- **이번 업데이트 이전에는 cluster-logging-operator 에서 클러스터 범위 역할 및 바인딩을 사용하여 지표를 스크랩하는 Prometheus 서비스 계정에 대한 권한을 설정했습니다. 이러한 권한은 콘솔 인터페이스를 사용하여 Operator를 배포할 때 생성되었지만 명령줄에서 배포할 때 누락되었습니다. 이번 업데이트에서는 roles 및 bindings 네임스페이스 범위를 설정하여 문제를 해결합니다. (LOG-2286)**
- **이번 업데이트 이전에는 수정 대시보드 조정에 대한 이전 변경 사항으로 네임스페이스 간 리소스에 ownerReferences 필드가 도입되었습니다. 결과적으로 네임스페이스에 구성 맵과 대시보드가 생성되지 않았습니다. 이번 업데이트를 통해 ownerReferences 필드가 제거되어 문제가 해결되고 콘솔에서 OpenShift Logging 대시보드를 사용할 수 있습니다. (LOG-2163)**

- 이번 업데이트 이전에는 **cluster-logging-operator** 가 대시보드가 포함된 기존 및 수정된 구성 맵과 올바르게 비교되지 않았기 때문에 메트릭 대시보드 변경 사항이 배포되지 않았습니다. 이번 업데이트를 통해 오브젝트 라벨에 고유한 해시 값을 추가하면 문제가 해결됩니다. (LOG-2071)

- 이번 업데이트 이전에는 OpenShift Logging 대시보드에 최근 24시간 동안 수집된 상위 생성 컨테이너가 표시되는 테이블에 Pod 및 네임스페이스가 올바르게 표시되지 않았습니다. 이번 업데이트를 통해 Pod 및 네임스페이스가 올바르게 표시됩니다. (LOG-2069)

- 이번 업데이트 이전에는 **Elasticsearch OutputDefault** 및 Elasticsearch 출력에 구조화된 키가 없는 **ClusterLogForwarder** 가 설정된 경우 생성된 구성에 인증에 대한 잘못된 값이 포함되어 있습니다. 이번 업데이트에서는 사용된 보안 및 인증서가 수정되었습니다. (LOG-2056)

- 이번 업데이트 이전에는 잘못된 메트릭에 대한 참조로 인해 OpenShift Logging 대시보드에 빈 CPU 그래프가 표시되었습니다. 이번 업데이트를 통해 올바른 데이터 지점을 선택하여 문제를 해결합니다. (LOG-2026)

- 이번 업데이트 이전에는 Fluentd 컨테이너 이미지에 런타임에 불필요한 빌더 툴이 포함되었습니다. 이번 업데이트에서는 이미지에서 이러한 도구를 제거합니다. (LOG-1927)

- 이번 업데이트 이전에는 5.3 릴리스에 배포된 수집기의 이름 변경으로 인해 로깅 수집기에서 **FluentdNodeDown** 경고가 생성되었습니다. 이번 업데이트에서는 Prometheus 경고의 작업 이름을 수정하여 문제를 해결합니다. (LOG-1918)

- 이번 업데이트 이전에는 구성 요소 이름의 리팩토링으로 인해 로그 수집기에서 자체 로그를 수집했습니다. 이로 인해 수집기의 피드백 루프가 메모리 및 로그 메시지 크기 문제가 발생할 수 있는 자체 로그를 처리할 수 있습니다.This lead to a potential feedback loop of the collector processing its own log that might result in memory and log message size issues. 이번 업데이트에서는 컬렉션에서 수집기 로그를 제외하여 문제를 해결합니다. (LOG-1774)

- 이번 업데이트 이전에는 PVC가 이미 존재하는 경우 **forbidden: forbidden: exceeded quota: infra-storage-quota.** 이번 업데이트를 통해 Elasticsearch는 기존 PVC를 확인하여 문제를 해결합니다. (LOG-2131)

- 이번 업데이트 이전에는 **elasticsearch-signing** secret이 제거되었을 때 Elasticsearch가 ready 상태로 돌아갈 수 없었습니다. 이번 업데이트를 통해 Elasticsearch는 해당 시크릿이 제거된 후 ready 상태로 돌아갈 수 있습니다. (LOG-2171)

- 이번 업데이트 이전에는 수집기에서 컨테이너 로그를 읽는 경로 변경으로 인해 수집기에서 일부 레코드를 잘못된 인덱스로 전달했습니다. 이번 업데이트를 통해 이제 수집기에서 올바른 구성을 사용하여 문제를 해결합니다. (LOG-2160)

- 이번 업데이트 이전에는 네임스페이스 수가 많은 클러스터에서 네임스페이스 목록에 최대 헤더 크기 제한에 도달했기 때문에 요청 제공이 중지되었습니다. 이번 업데이트를 통해 헤더에는 네임스페이스 이름 목록만 포함되어 문제를 해결합니다. (LOG-1899)

- 이번 업데이트 이전에는 OpenShift Container Platform 로깅 대시보드에 Elasticsearch의 노드가 'x'인 경우 shard 'x' 횟수가 실제 값보다 큰 수가 표시되었습니다. 이 문제는 각 Elasticsearch Pod에 대한 모든 기본 shard를 출력하고 전체 Elasticsearch 클러스터에 대한 출력이 항상 전송되었지만 이로 인한 합계를 계산하기 때문에 발생했습니다. 이번 업데이트를 통해 이제 shard 수가 올바르게 계산됩니다. (LOG-2156)

- 이번 업데이트 이전에는 수동으로 삭제된 경우 secrets **kibana** 및 **kibana-proxy** 가 다시 생성되지 않았습니다. 이번 업데이트를 통해 **elasticsearch-operator** 는 리소스를 감시하고 삭제된 경우 자동으로 다시 생성합니다. (LOG-2250)

- 이번 업데이트 이전에는 버퍼 청크 크기를 튜닝하면 수집기에서 이벤트 스트림의 바이트 제한을 초과하는 청크 크기에 대한 경고를 생성할 수 있습니다. 이번 업데이트를 통해 읽기 행 제한을 조정하여 문제를 해결할 수도 있습니다. (LOG-2379)
- 이번 업데이트 이전에는 OpenShift 웹 콘솔의 로깅 콘솔 링크가 ClusterLogging CR에서 제거되지 않았습니다. 이번 업데이트를 통해 CR을 삭제하거나 Cluster Logging Operator를 제거하면 링크가 제거됩니다. (LOG-2373)
- 이번 업데이트 이전에는 컨테이너 로그 경로를 변경하면 원래 경로로 구성된 이전 릴리스와 함께 collection 메트릭이 항상 0이 되었습니다. 이번 업데이트를 통해 수집된 로그에 대한 지표를 표시하는 플러그인은 문제를 해결하기 위해 두 경로에서 읽기를 지원합니다. (LOG-2462)

1.9.5. CVE

- CVE-2022-0759
  - BZ-2058404
- CVE-2022-21698
  - BZ-2045880

1.10. LOGGING 5.3.13

이번 릴리스에는 RHSA-2022:68828-OpenShift Logging 버그 수정 릴리스 5.3.13이 포함되어 있습니다.

1.10.1. 버그 수정

- 이번 업데이트 이전에는 **log-file-metrics-exporter** 구성 요소에서 생성한 로그 파일 크기 맵에서 삭제된 파일의 항목을 제거하지 않아 파일 크기가 증가되고 메모리가 처리되었습니다. 이번 업데이트를 통해 로그 파일 크기 맵에 삭제된 파일에 대한 항목이 포함되어 있지 않습니다. (LOG-3293)

1.10.2. CVE

예 1.4. CVE를 확장하려면 클릭합니다.

- 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-2022-39399
- CVE-2022-40674

1.11. LOGGING 5.3.12

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.3.12 가 포함되어 있습니다.

1.11.1. 버그 수정

없음.

1.11.2. 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

1.12. LOGGING 5.3.11

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.3.11가 포함되어 있습니다.

1.12.1. 버그 수정

- 이번 업데이트 이전에는 Operator에서 Pod가 준비되었는지 확인하지 않아 클러스터를 다시 시작하는 동안 클러스터가 작동하지 않는 상태에 도달했습니다. 이번 업데이트를 통해 Operator는 재시작 중에 새 Pod를 계속 진행하기 전에 새 Pod를 준비 상태로 표시하여 문제를 해결합니다. (LOG-2871)

1.12.2. CVE

- CVE-2022-1292
- CVE-2022-1586
- CVE-2022-1785
- CVE-2022-1897
- CVE-2022-1927
- CVE-2022-2068
- CVE-2022-2097
- CVE-2022-30631

1.13. LOGGING 5.3.10

이 릴리스에는 RHSA-2022:5908-OpenShift Logging 버그 수정 릴리스 5.3.10이 포함되어 있습니다.

1.13.1. 버그 수정

- BZ-2100495

1.13.2. CVE

> 예 1.5. CVE를 확장하려면 클릭합니다.
>
> - CVE-2021-38561
> - CVE-2021-40528
> - CVE-2022-1271
> - CVE-2022-1621
> - CVE-2022-1629
> - CVE-2022-21540
> - CVE-2022-21541
> - CVE-2022-22576
> - CVE-2022-25313
> - CVE-2022-25314
> - CVE-2022-27774

- CVE-2022-27776
- CVE-2022-27782
- CVE-2022-29824
- CVE-2022-34169

1.14. 로깅 5.3.9

이 릴리스에는 RHBA-2022:5557-OpenShift Logging 버그 수정 릴리스 5.3.9가 포함되어 있습니다.

1.14.1. 버그 수정

- 이번 업데이트 이전에는 로깅 수집기가 생성한 지표의 레이블로 경로를 포함했습니다. 이 경로는 자주 변경되어 Prometheus 서버의 중요한 스토리지 변경에 기여했습니다. 이번 업데이트를 통해 문제를 해결하고 스토리지 소비를 줄이기 위해 레이블이 삭제되었습니다. (LOG-2682)

1.14.2. CVE

예 1.6. CVE를 확장하려면 클릭합니다.

- CVE-2020-28915
- CVE-2021-40528
- CVE-2022-1271
- CVE-2022-1621
- CVE-2022-1629
- CVE-2022-22576
- CVE-2022-25313
- CVE-2022-25314
- CVE-2022-26691
- CVE-2022-27666
- CVE-2022-27774
- CVE-2022-27776
- CVE-2022-27782
- CVE-2022-29824

1.15. LOGGING 5.3.8

이 릴리스에는 RHBA-2022:5010-OpenShift Logging 버그 수정 릴리스 5.3.8이 포함되어 있습니다.

1.15.1. 버그 수정

(none.)

1.15.2. CVE

예 1.7. CVE를 확장하려면 클릭합니다.

- CVE-2018-25032
- CVE-2020-0404
- CVE-2020-4788
- CVE-2020-13974
- CVE-2020-19131
- CVE-2020-27820
- CVE-2021-0941
- CVE-2021-3612
- CVE-2021-3634
- CVE-2021-3669
- CVE-2021-3737
- CVE-2021-3743
- CVE-2021-3744
- CVE-2021-3752
- CVE-2021-3759
- CVE-2021-3764
- CVE-2021-3772
- CVE-2021-3773
- CVE-2021-4002
- CVE-2021-4037
- CVE-2021-4083
- CVE-2021-4157
- CVE-2021-4189
- CVE-2021-4197
- CVE-2021-4203

- CVE-2021-20322
- CVE-2021-21781
- CVE-2021-23222
- CVE-2021-26401
- CVE-2021-29154
- CVE-2021-37159
- CVE-2021-41617
- CVE-2021-41864
- CVE-2021-42739
- CVE-2021-43056
- CVE-2021-43389
- CVE-2021-43976
- CVE-2021-44733
- CVE-2021-45485
- CVE-2021-45486
- CVE-2022-0001
- CVE-2022-0002
- CVE-2022-0286
- CVE-2022-0322
- CVE-2022-1011
- CVE-2022-1271

1.16. OPENSHIFT LOGGING 5.3.7

이번 릴리스에는 RHSA-2022:2217 OpenShift Logging 버그 수정 릴리스 5.3.7이 포함되어 있습니다.

1.16.1. 버그 수정

- 이번 업데이트 이전에는 Linux 감사 로그 시간 구문 분석이 키/값 쌍의 직사 위치에 의존했습니다. 이번 업데이트에서는 정규식을 변경하여 시간 항목을 찾습니다. (LOG-2322)
- 이번 업데이트 이전에는 일부 로그 전달자 출력이 동일한 타임 스탬프를 사용하여 로그를 다시 정렬할 수 있습니다. 이번 업데이트를 통해 타임스탬프와 일치하는 항목을 정렬하기 위해 로그 레코드에 시퀀스 번호가 추가되었습니다. (LOG-2334)

- 이번 업데이트 이전에는 네임스페이스 수가 많은 클러스터에서 네임스페이스 목록에 최대 헤더 크기 제한에 도달했기 때문에 요청 제공이 중지되었습니다. 이번 업데이트를 통해 헤더에는 네임스페이스 이름 목록만 포함되어 문제를 해결합니다. (LOG-2450)

- 이번 업데이트 이전에는 **system:serviceaccount:openshift-monitoring:prometheus-k8s** 에 clusterrole 및 **cluster role binding** 로 클러스터 수준 권한이 있었습니다. 이번 업데이트에서는 **serviceaccount** 를 역할 및 rolebinding을 사용하여**openshift-logging** 네임스페이스로 제한합니다. (LOG-2481))

1.16.2. CVE

예 1.8. CVE를 확장하려면 클릭합니다.

- CVE-2018-25032
- CVE-2021-4028
- CVE-2021-37136
- CVE-2021-37137
- CVE-2021-43797
- CVE-2022-0759
- CVE-2022-0778
- CVE-2022-1154
- CVE-2022-1271
- CVE-2022-21426
- CVE-2022-21434
- CVE-2022-21443
- CVE-2022-21476
- CVE-2022-21496
- CVE-2022-21698
- CVE-2022-25636

1.17. OPENSHIFT LOGGING 5.3.6

이번 릴리스에는 RHBA-2022:1377 OpenShift Logging 버그 수정 릴리스 5.3.6이 포함되어 있습니다.

1.17.1. 버그 수정

- 이번 업데이트 이전에는 키가 없는 허용 오차를 정의하고 기존 Operator를 정의하면 Operator에서 업그레이드를 완료할 수 없었습니다. 이번 업데이트를 통해 이 허용 오차가 더 이상 업그레이드 완료를 차단하지 않습니다. (LOG-2126)

- 이 변경 이전에는 수집기에서 청크 바이트 제한이 내보낸 이벤트를 초과하는 경고를 생성할 수 있었습니다. 이번 변경으로 readline 제한을 조정하여 업스트림 문서에서 권장하는 문제를 해결할 수 있습니다. (LOG-2380)

1.18. OPENSHIFT LOGGING 5.3.5

이 릴리스에는 RHSA-2022:0721 OpenShift Logging 버그 수정 릴리스 5.3.5가 포함되어 있습니다.

1.18.1. 버그 수정

- 이번 업데이트 이전에는 OpenShift Container Platform에서 OpenShift Logging을 제거한 경우 웹 콘솔에서 로깅 페이지에 대한 링크를 계속 표시합니다. 이번 업데이트를 통해 OpenShift Logging을 제거하거나 제거하면 해당 링크도 제거됩니다. (LOG-2182)

1.18.2. CVE

예 1.9. CVE를 확장하려면 클릭합니다.

- CVE-2020-28491
- CVE-2021-3521
- CVE-2021-3872
- CVE-2021-3984
- CVE-2021-4019
- CVE-2021-4122
- CVE-2021-4192
- CVE-2021-4193
- CVE-2022-0552

1.19. OPENSHIFT LOGGING 5.3.4

이번 릴리스에는 RHBA-2022:0411 OpenShift Logging 버그 수정 릴리스 5.3.4가 포함되어 있습니다.

1.19.1. 버그 수정

- 이번 업데이트 이전에는 **cluster-logging-operator** 가 대시보드가 포함된 기존 및 원하는 구성 맵을 올바르게 비교하지 않았기 때문에 메트릭 대시보드 변경 사항이 아직 배포되지 않았습니다. 이번 업데이트에서는 오브젝트 라벨에 고유한 해시 값을 추가하여 논리를 수정합니다. (LOG-2066)
- 이번 업데이트 이전에는 FIPS가 활성화된 후 Elasticsearch Pod를 시작하지 못했습니다. 이번 업데이트를 통해 Elasticsearch Pod가 성공적으로 시작됩니다. (LOG-1974)
- 이번 업데이트 이전에는 PVC가 이미 존재하는 경우 forbidden: forbidden: exceeded quota: infra-storage-quota."라는 오류를 "Unable to create PersistentVolumeClaim을 생성할 수 없습니다. 이번 업데이트를 통해 elasticsearch는 기존 PVC를 확인하여 문제를 해결합니다. (LOG-2127)

1.19.2. CVE

예 1.10. CVE를 확장하려면 클릭합니다.

- CVE-2021-3521
- CVE-2021-3872
- CVE-2021-3984
- CVE-2021-4019
- CVE-2021-4122
- CVE-2021-4155
- CVE-2021-4192
- CVE-2021-4193
- CVE-2022-0185
- CVE-2022-21248
- CVE-2022-21277
- CVE-2022-21282
- CVE-2022-21283
- CVE-2022-21291
- CVE-2022-21293
- CVE-2022-21294
- CVE-2022-21296
- CVE-2022-21299
- CVE-2022-21305
- CVE-2022-21340
- CVE-2022-21341
- CVE-2022-21360
- CVE-2022-21365
- CVE-2022-21366

1.20. OPENSHIFT LOGGING 5.3.3

이번 릴리스에는 RHSA-2022:0227 OpenShift Logging 버그 수정 릴리스 5.3.3이 포함되어 있습니다.

1.20.1. 버그 수정

- 이번 업데이트 이전에는 cluster-logging-operator가 대시보드가 포함된 기존 및 원하는 configmaps를 올바르게 비교하지 않았기 때문에 메트릭 대시보드 변경 사항이 아직 배포되지 않았습니다. 이번 업데이트에서는 대시보드 고유 해시 값을 오브젝트 라벨에 추가하여 논리를 수정합니다. (LOG-2066)

- 이번 업데이트에서는 CVE-2021-44832를 해결하기 위해 log4j 종속성을 2.17.1로 변경하여 CVE-2021-44832.(LOG-2102)를 해결합니다.

1.20.2. CVE

> 예 1.11. CVE를 확장하려면 클릭합니다.
>
> - CVE-2021-27292
>   - BZ-1940613
> - CVE-2021-44832
>   - BZ-2035951

1.21. OPENSHIFT LOGGING 5.3.2

이번 릴리스에는 RHSA-2022:0044 OpenShift Logging 버그 수정 릴리스 5.3.2가 포함되어 있습니다.

1.21.1. 버그 수정

- 이번 업데이트 이전에는 구문 분석 오류로 인해 Elasticsearch가 이벤트 라우터에서 로그를 거부했습니다. 이번 업데이트에서는 구문 분석 오류를 해결하기 위해 데이터 모델을 변경합니다. 그러나 이전 인덱스로 인해 Kibana 내에서 경고 또는 오류가 발생할 수 있습니다. **kubernetes.event.metadata.resourceVersion** 필드는 기존 인덱스가 제거되거나 다시 인덱싱될 때까지 오류를 발생시킵니다. 이 필드를 Kibana에서 사용하지 않으면 오류 메시지를 무시할 수 있습니다. 이전 인덱스를 삭제하는 보존 정책이 있는 경우 정책은 결국 이전 인덱스를 제거하고 오류 메시지를 중지합니다. 그렇지 않으면 오류 메시지를 수동으로 다시 시작하여 오류 메시지를 중지합니다. (LOG-2087)

- 이번 업데이트 이전에는 OpenShift Logging 대시보드에 지난 24시간 동안 최상위 생성 및 수집된 컨테이너를 표시하는 테이블에 잘못된 Pod 네임스페이스가 표시되었습니다. 이번 업데이트를 통해 OpenShift Logging 대시보드에 올바른 Pod 네임스페이스가 표시됩니다. (LOG-2051)

- 이번 업데이트 이전에는 **ClusterLogForwarder** 사용자 정의 리소스(CR) 인스턴스의 **outputDefaults.elasticsearch.structuredTypeKey** 에 구조화된 키가 없는 경우 CR에서 출력 시크릿을 기본 로그 저장소와 통신하는 데 사용되는 기본 시크릿으로 교체했습니다. 이번 업데이트를 통해 정의된 출력 시크릿이 올바르게 사용됩니다. (LOG-2046)

1.21.2. CVE

> 예 1.12. CVE를 확장하려면 클릭합니다.
>
> - CVE-2020-36327
>   - BZ-1958999

- CVE-2021-45105
  - BZ-2034067
- CVE-2021-3712
- CVE-2021-20321
- CVE-2021-42574

1.22. OPENSHIFT LOGGING 5.3.1

이번 릴리스에는 RHSA-2021:5129 OpenShift Logging 버그 수정 릴리스 5.3.1이 포함되어 있습니다.

1.22.1. 버그 수정

- 이번 업데이트 이전에는 Fluentd 컨테이너 이미지에 런타임에 불필요한 빌더 툴이 포함되었습니다. 이번 업데이트에서는 이미지에서 이러한 도구를 제거합니다. (LOG-1998)
- 이번 업데이트 이전에는 잘못된 메트릭에 대한 참조로 인해 로깅 대시보드에 빈 CPU 그래프가 표시되었습니다. 이번 업데이트를 통해 로깅 대시보드에 CPU 그래프가 올바르게 표시됩니다. (LOG-1925)
- 이번 업데이트 이전에는 Elasticsearch 노드 성능에 영향을 미치는 고비용 쿼리를 사용하여 Elasticsearch Prometheus 내보내기 플러그인의 인덱스 수준 지표를 컴파일했습니다. 이번 업데이트에서는 성능이 향상되는 저렴한 쿼리를 구현합니다. (LOG-1897)

1.22.2. CVE

예 1.13. CVE를 확장하려면 클릭합니다.

- CVE-2021-21409
  - BZ-1944888
- CVE-2021-37136
  - BZ-2004133
- CVE-2021-37137
  - BZ-2004135
- CVE-2021-44228
  - BZ-2030932
- CVE-2018-25009
- CVE-2018-25010
- CVE-2018-25012
- CVE-2018-25013

- CVE-2018-25014
- CVE-2019-5827
- CVE-2019-13750
- CVE-2019-13751
- CVE-2019-17594
- CVE-2019-17595
- CVE-2019-18218
- CVE-2019-19603
- CVE-2019-20838
- CVE-2020-12762
- CVE-2020-13435
- CVE-2020-14145
- CVE-2020-14155
- CVE-2020-16135
- CVE-2020-17541
- CVE-2020-24370
- CVE-2020-35521
- CVE-2020-35522
- CVE-2020-35523
- CVE-2020-35524
- CVE-2020-36330
- CVE-2020-36331
- CVE-2020-36332
- CVE-2021-3200
- CVE-2021-3426
- CVE-2021-3445
- CVE-2021-3481
- CVE-2021-3572
- CVE-2021-3580

- CVE-2021-3712
- CVE-2021-3800
- CVE-2021-20231
- CVE-2021-20232
- CVE-2021-20266
- CVE-2021-20317
- CVE-2021-22876
- CVE-2021-22898
- CVE-2021-22925
- CVE-2021-27645
- CVE-2021-28153
- CVE-2021-31535
- CVE-2021-33560
- CVE-2021-33574
- CVE-2021-35942
- CVE-2021-36084
- CVE-2021-36085
- CVE-2021-36086
- CVE-2021-36087
- CVE-2021-42574
- CVE-2021-43267
- CVE-2021-43527
- CVE-2021-45046

1.23. OPENSHIFT LOGGING 5.3.0

이번 릴리스에는 RHSA-2021:4627 OpenShift Logging 버그 수정 릴리스 5.3.0이 포함되어 있습니다.

1.23.1. 새로운 기능 및 개선 사항

- 이번 업데이트를 통해 로그 전달에 대한 권한 부여 옵션이 확장되었습니다. 이제 SASL, 사용자 이름/암호 또는 TLS를 사용하여 출력을 구성할 수 있습니다.

1.23.2. 버그 수정

- 이번 업데이트 이전에는 syslog 프로토콜을 사용하여 로그를 전달한 경우 ruby 해시로 인코딩된 키/값 쌍을 직렬화하여 'ECDHE' 문자와 "#11" 탭을 교체했습니다. 이번 업데이트에서는 로그 메시지가 유효한 JSON으로 올바르게 직렬화되도록 문제가 해결되었습니다. (LOG-1494)

- 이번 업데이트 이전에는 여러 줄 오류 탐지가 활성화된 상태에서 적절한 Cloudwatch 스트림으로 전달하도록 애플리케이션 로그를 올바르게 구성하지 않았습니다. (LOG-1939)

- 이번 업데이트 이전에는 5.3 릴리스의 배포된 수집기의 이름 변경으로 인해 경고 'fluentnodedown'이 생성되었습니다. (LOG-1918)

- 이번 업데이트 이전에는 이전 릴리스 구성에 발생한 회귀 문제로 인해 수집기에서 종료 전에 버퍼링된 메시지를 플러시하여 종료 후 수집기 Pod를 다시 시작합니다. 이번 업데이트를 통해 fluentd는 더 이상 종료 시 버퍼를 플러시하지 않아 문제를 해결합니다. (LOG-1735)

- 이번 업데이트 이전에는 이전 릴리스에서 의도적으로 비활성화된 JSON 메시지 구문 분석이 도입된 회귀 문제가 있었습니다. 이번 업데이트에서는 JSON 구문 분석을 다시 활성화합니다. 또한 구문 분석된 JSON 메시지의 "level" 필드를 기반으로 로그 항목 "level"을 설정하거나 regex를 사용하여 메시지 필드에서 일치 항목을 추출합니다. (LOG-1199)

- 이번 업데이트 이전에는 필요한 버퍼 공간을 사용할 수 없는 경우에도 **ClusterLogging** 사용자 정의 리소스(CR)에서 **totalLimitSize** 필드의 값을 Fluentd**total_limit_size** 필드에 적용했습니다. 이번 업데이트를 통해 CR은 두 개의 **totalLimitSize** 또는 'default' 값 중 더 적은 값을 Fluentd **total_limit_size** 필드에 적용하여 문제를 해결합니다. (LOG-1776)

1.23.3. 확인된 문제

- 외부 Elasticsearch 서버로 로그를 전달한 다음 사용자 이름 및 암호와 같은 파이프라인 시크릿에 구성된 값을 변경하는 경우 Fluentd 전달자가 새 시크릿을 로드하지만 이전 값을 사용하여 외부 Elasticsearch 서버에 연결합니다. 이 문제는 Red Hat OpenShift Logging Operator가 현재 콘텐츠 변경 사항에 대한 시크릿을 모니터링하지 않기 때문에 발생합니다. (LOG-1652)
  문제를 해결하기 위해 시크릿을 변경하는 경우 다음을 입력하여 Fluentd Pod를 강제로 재배포할 수 있습니다.

  ```
  $ oc delete pod -l component=collector
  ```

1.23.4. 사용되지 않거나 삭제된 기능

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

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

1.23.4.1. 레거시 Fluentd 및 레거시 syslog 방법을 사용하여 로그 전달이 제거됨

OpenShift Logging 5.3에서는 기존 로그를 Syslog 및 Fluentd로 전달하는 방법이 제거됩니다. 버그 수정 및 지원은 OpenShift Logging 5.2 라이프 사이클 종료를 통해 제공됩니다. 그 이후에는 새로운 기능 개선이 이루어지지 않습니다.

대신 다음과 같은 비레거시 방법을 사용합니다.

- Fluentd 정방향 프로토콜을 사용하여 로그 전달

- syslog 프로토콜을 사용하여 로그 전달

1.23.4.2. 레거시 전달 방법에 대한 구성 메커니즘이 제거되었습니다.

OpenShift Logging 5.3에서 로그 전달을 위한 기존 구성 메커니즘이 제거되었습니다. 레거시 Fluentd 방법 및 기존 Syslog 방법을 사용하여 로그를 전달할 수 없습니다. 대신 표준 로그 전달 방법을 사용합니다.

1.23.5. CVE

예 1.14. CVE를 확장하려면 클릭합니다.

- CVE-2018-20673
- CVE-2018-25009
- CVE-2018-25010
- CVE-2018-25012
- CVE-2018-25013
- CVE-2018-25014
- CVE-2019-5827
- CVE-2019-13750
- CVE-2019-13751
- CVE-2019-14615
- CVE-2019-17594
- CVE-2019-17595
- CVE-2019-18218
- CVE-2019-19603
- CVE-2019-20838
- CVE-2020-0427
- CVE-2020-10001
- CVE-2020-12762
- CVE-2020-13435
- CVE-2020-14145
- CVE-2020-14155
- CVE-2020-16135
- CVE-2020-17541
- CVE-2020-24370

- CVE-2020-24502
- CVE-2020-24503
- CVE-2020-24504
- CVE-2020-24586
- CVE-2020-24587
- CVE-2020-24588
- CVE-2020-26139
- CVE-2020-26140
- CVE-2020-26141
- CVE-2020-26143
- CVE-2020-26144
- CVE-2020-26145
- CVE-2020-26146
- CVE-2020-26147
- CVE-2020-27777
- CVE-2020-29368
- CVE-2020-29660
- CVE-2020-35448
- CVE-2020-35521
- CVE-2020-35522
- CVE-2020-35523
- CVE-2020-35524
- CVE-2020-36158
- CVE-2020-36312
- CVE-2020-36330
- CVE-2020-36331
- CVE-2020-36332
- CVE-2020-36386
- CVE-2021-0129

- CVE-2021-3200
- CVE-2021-3348
- CVE-2021-3426
- CVE-2021-3445
- CVE-2021-3481
- CVE-2021-3487
- CVE-2021-3489
- CVE-2021-3564
- CVE-2021-3572
- CVE-2021-3573
- CVE-2021-3580
- CVE-2021-3600
- CVE-2021-3635
- CVE-2021-3659
- CVE-2021-3679
- CVE-2021-3732
- CVE-2021-3778
- CVE-2021-3796
- CVE-2021-3800
- CVE-2021-20194
- CVE-2021-20197
- CVE-2021-20231
- CVE-2021-20232
- CVE-2021-20239
- CVE-2021-20266
- CVE-2021-20284
- CVE-2021-22876
- CVE-2021-22898
- CVE-2021-22925

- CVE-2021-23133
- CVE-2021-23840
- CVE-2021-23841
- CVE-2021-27645
- CVE-2021-28153
- CVE-2021-28950
- CVE-2021-28971
- CVE-2021-29155
- ICVE-2021-29646
- CVE-2021-29650
- CVE-2021-31440
- CVE-2021-31535
- CVE-2021-31829
- CVE-2021-31916
- CVE-2021-33033
- CVE-2021-33194
- CVE-2021-33200
- CVE-2021-33560
- CVE-2021-33574
- CVE-2021-35942
- CVE-2021-36084
- CVE-2021-36085
- CVE-2021-36086
- CVE-2021-36087
- CVE-2021-42574

1.24. LOGGING 5.2.13

이번 릴리스에는 RHSA-2022:5909-OpenShift Logging 버그 수정 릴리스 5.2.13 이 포함되어 있습니다.

1.24.1. 버그 수정

- BZ-2100495

1.24.2. CVE

예 1.15. CVE를 확장하려면 클릭합니다.

- CVE-2021-38561
- CVE-2021-40528
- CVE-2022-1271
- CVE-2022-1621
- CVE-2022-1629
- CVE-2022-21540
- CVE-2022-21541
- CVE-2022-22576
- CVE-2022-25313
- CVE-2022-25314
- CVE-2022-27774
- CVE-2022-27776
- CVE-2022-27782
- CVE-2022-29824
- CVE-2022-34169

1.25. LOGGING 5.2.12

이 릴리스에는 RHBA-2022:5558-OpenShift Logging 버그 수정 릴리스 5.2.12이 포함되어 있습니다.

1.25.1. 버그 수정

없음.

1.25.2. CVE

예 1.16. CVE를 확장하려면 클릭합니다.

- CVE-2020-28915
- CVE-2021-40528
- CVE-2022-1271

- CVE-2022-1621
- CVE-2022-1629
- CVE-2022-22576
- CVE-2022-25313
- CVE-2022-25314
- CVE-2022-26691
- CVE-2022-27666
- CVE-2022-27774
- CVE-2022-27776
- CVE-2022-27782
- CVE-2022-29824

1.26. LOGGING 5.2.11

이 릴리스에는 RHBA-2022:5012-OpenShift Logging 버그 수정 릴리스 5.2.11이 포함되어 있습니다.

1.26.1. 버그 수정

- 이번 업데이트 이전에는 사용자가 거부된 로그 파일을 임시 스토리지로 전송하도록 구성된 클러스터에서 시간이 지남에 따라 클러스터 불안정성을 초래합니다. 이번 업데이트를 통해 Gradle의 청크 백업이 비활성화되어 문제를 해결합니다. (LOG-2635)

1.26.2. CVE

예 1.17. CVE를 확장하려면 클릭합니다.

- CVE-2018-25032
- CVE-2020-0404
- CVE-2020-4788
- CVE-2020-13974
- CVE-2020-19131
- CVE-2020-27820
- CVE-2021-0941
- CVE-2021-3612
- CVE-2021-3634
- CVE-2021-3669

- CVE-2021-3737
- CVE-2021-3743
- CVE-2021-3744
- CVE-2021-3752
- CVE-2021-3759
- CVE-2021-3764
- CVE-2021-3772
- CVE-2021-3773
- CVE-2021-4002
- CVE-2021-4037
- CVE-2021-4083
- CVE-2021-4157
- CVE-2021-4189
- CVE-2021-4197
- CVE-2021-4203
- CVE-2021-20322
- CVE-2021-21781
- CVE-2021-23222
- CVE-2021-26401
- CVE-2021-29154
- CVE-2021-37159
- CVE-2021-41617
- CVE-2021-41864
- CVE-2021-42739
- CVE-2021-43056
- CVE-2021-43389
- CVE-2021-43976
- CVE-2021-44733
- CVE-2021-45485

- CVE-2021-45486
- CVE-2022-0001
- CVE-2022-0002
- CVE-2022-0286
- CVE-2022-0322
- CVE-2022-1011
- CVE-2022-1271

1.27. OPENSHIFT LOGGING 5.2.10

이 릴리스에는 OpenShift Logging 버그 수정 릴리스 5.2.10]이 포함되어 있습니다.

1.27.1. 버그 수정

- 이번 업데이트 이전에는 일부 로그 전달자 출력이 동일한 타임 스탬프를 사용하여 로그를 다시 정렬할 수 있습니다. 이번 업데이트를 통해 타임스탬프와 일치하는 항목을 정렬하기 위해 로그 레코드에 시퀀스 번호가 추가되었습니다.(LOG-2335)
- 이번 업데이트 이전에는 네임스페이스 수가 많은 클러스터에서 네임스페이스 목록에 최대 헤더 크기 제한에 도달했기 때문에 요청 제공이 중지되었습니다. 이번 업데이트를 통해 헤더에는 네임스페이스 이름 목록만 포함되어 문제를 해결합니다. (LOG-2475)
- 이번 업데이트 이전에는 **system:serviceaccount:openshift-monitoring:prometheus-k8s** 에 clusterrole 및 **cluster role binding** 로 클러스터 수준 권한이 있었습니다. 이번 업데이트에서는 **serviceaccount** 를 역할 및 rolebinding을 사용하여**openshift-logging** 네임스페이스로 제한합니다. (LOG-2480)
- 이번 업데이트 이전에는 **cluster-logging-operator** 에서 클러스터 범위 역할 및 바인딩을 활용하여 Prometheus 서비스 계정에서 지표를 스크랩할 수 있는 권한을 설정합니다. 이러한 권한은 콘솔 인터페이스를 사용하여 Operator를 배포하고 명령줄에서 Operator를 배포할 때 누락된 경우에만 생성되었습니다. 이렇게 하면 이 역할 및 바인딩 네임스페이스 범위가 지정되어 문제가 해결되었습니다. (LOG-1972)

1.27.2. CVE

예 1.18. CVE를 확장하려면 클릭합니다.

- CVE-2018-25032
- CVE-2021-4028
- CVE-2021-37136
- CVE-2021-37137
- CVE-2021-43797
- CVE-2022-0778

- CVE-2022-1154
- CVE-2022-1271
- CVE-2022-21426
- CVE-2022-21434
- CVE-2022-21443
- CVE-2022-21476
- CVE-2022-21496
- CVE-2022-21698
- CVE-2022-25636

1.28. OPENSHIFT LOGGING 5.2.9

이번 릴리스에는 RHBA-2022:1375 OpenShift Logging 버그 수정 릴리스 5.2.9]가 포함되어 있습니다.

1.28.1. 버그 수정

- 이번 업데이트 이전에는 키가 없는 허용 오차를 정의하고 기존 Operator를 정의하면 Operator에서 업그레이드를 완료할 수 없었습니다. 이번 업데이트를 통해 이 허용 오차가 더 이상 업그레이드 완료를 차단하지 않습니다. (LOG-2304)

1.29. OPENSHIFT LOGGING 5.2.8

이번 릴리스에는 RHSA-2022:0728 OpenShift Logging 버그 수정 릴리스 5.2.8이 포함되어 있습니다.

1.29.1. 버그 수정

- 이번 업데이트 이전에는 OpenShift Container Platform에서 OpenShift Logging을 제거한 경우 웹 콘솔에서 로깅 페이지에 대한 링크를 계속 표시합니다. 이번 업데이트를 통해 OpenShift Logging을 제거하거나 제거하면 해당 링크도 제거됩니다. (LOG-2180)

1.29.2. CVE

예 1.19. CVE를 확장하려면 클릭합니다.

- CVE-2020-28491
  - BZ-1930423
- CVE-2022-0552
  - BG-2052539

1.30. OPENSHIFT LOGGING 5.2.7

이번 릴리스에는 RHBA-2022:0478 OpenShift Logging 버그 수정 릴리스 5.2.7이 포함되어 있습니다.

1.30.1. 버그 수정

- 이번 업데이트 이전에는 FIPS가 활성화된 Elasticsearch Pod가 업데이트 후 시작되지 않았습니다. 이번 업데이트를 통해 Elasticsearch Pod가 성공적으로 시작됩니다. (LOG-2000)
- 이번 업데이트 이전에는 PVC(영구 볼륨 클레임)가 이미 존재하는 경우 Elasticsearch에서 금지된 quota: infra-storage-quota"로 인해 " PersistentVolumeClaim을 생성할 수 없음"이라는 오류가 발생했습니다. 이번 업데이트를 통해 Elasticsearch는 기존 PVC를 확인하여 문제를 해결합니다. (LOG-2118)

1.30.2. CVE

> 예 1.20. CVE를 확장하려면 클릭합니다.
>
> - CVE-2021-3521
> - CVE-2021-3872
> - CVE-2021-3984
> - CVE-2021-4019
> - CVE-2021-4122
> - CVE-2021-4155
> - CVE-2021-4192
> - CVE-2021-4193
> - CVE-2022-0185

1.31. OPENSHIFT LOGGING 5.2.6

이 릴리스에는 RHSA-2022:0230 OpenShift Logging 버그 수정 릴리스 5.2.6이 포함되어 있습니다.

1.31.1. 버그 수정

- 이번 업데이트 이전에는 릴리스에 Fluentd가 충돌한 필터 변경이 포함되지 않았습니다. 이번 업데이트를 통해 누락된 필터가 수정되었습니다. (LOG-2104)
- 이번 업데이트에서는 CVE-2021-44832를 해결하기 위해 log4j 종속성을 2.17.1로 변경하여 CVE-2021-44832.(LOG-2101)를 해결합니다.

1.31.2. CVE

> 예 1.21. CVE를 확장하려면 클릭합니다.
>
> - CVE-2021-27292
>   - BZ-1940613

- CVE-2021-44832
  - BZ-2035951

1.32. OPENSHIFT LOGGING 5.2.5

이 릴리스에는 RHSA-2022:0043 OpenShift Logging 버그 수정 릴리스 5.2.5가 포함되어 있습니다.

1.32.1. 버그 수정

- 이번 업데이트 이전에는 구문 분석 오류로 인해 Elasticsearch가 이벤트 라우터에서 로그를 거부했습니다. 이번 업데이트에서는 구문 분석 오류를 해결하기 위해 데이터 모델을 변경합니다. 그러나 이전 인덱스로 인해 Kibana 내에서 경고 또는 오류가 발생할 수 있습니다. **kubernetes.event.metadata.resourceVersion** 필드는 기존 인덱스가 제거되거나 다시 인덱싱될 때까지 오류를 발생시킵니다. 이 필드를 Kibana에서 사용하지 않으면 오류 메시지를 무시할 수 있습니다. 이전 인덱스를 삭제하는 보존 정책이 있는 경우 정책은 결국 이전 인덱스를 제거하고 오류 메시지를 중지합니다. 그렇지 않으면 오류 메시지를 수동으로 다시 시작하여 오류 메시지를 중지합니다. LOG-2087)

1.32.2. CVE

예 1.22. CVE를 확장하려면 클릭합니다.

- CVE-2021-3712
- CVE-2021-20321
- CVE-2021-42574
- CVE-2021-45105

1.33. OPENSHIFT LOGGING 5.2.4

이번 릴리스에는 RHSA-2021:5127 OpenShift Logging 버그 수정 릴리스 5.2.4가 포함되어 있습니다.

1.33.1. 버그 수정

- 이번 업데이트 이전에는 syslog를 통해 제공되는 레코드가 ruby 해시 인코딩 키/값 쌍을 직렬화하여 'Havana' 문자를 포함시키고, 탭을 "#11"로 바꿉니다. 이번 업데이트에서는 메시지가 적절한 JSON으로 올바르게 직렬화됩니다. (LOG-1775)

- 이번 업데이트 이전에는 Elasticsearch 노드 성능에 영향을 미치는 고비용 쿼리를 사용하여 Elasticsearch Prometheus 내보내기 플러그인의 인덱스 수준 지표를 컴파일했습니다. 이번 업데이트에서는 성능이 향상되는 저렴한 쿼리를 구현합니다. (LOG-1970)

- 이번 업데이트 이전에는 Log Forwarding이 여러 출력으로 구성된 경우 Elasticsearch가 거부된 경우가 있었습니다. 이는 수정된 출력 메시지 콘텐츠 중 하나를 단일 메시지로 구성하기 때문에 발생했습니다. 이번 업데이트를 통해 로그 전달은 출력별 처리가 다른 출력에 영향을 미치지 않도록 각 출력에 대한 메시지를 복제합니다. (LOG-1824)

1.33.2. CVE

예 1.23. CVE를 확장하려면 클릭합니다.

- CVE-2018-25009
- CVE-2018-25010
- CVE-2018-25012
- CVE-2018-25013
- CVE-2018-25014
- CVE-2019-5827
- CVE-2019-13750
- CVE-2019-13751
- CVE-2019-17594
- CVE-2019-17595
- CVE-2019-18218
- CVE-2019-19603
- CVE-2019-20838
- CVE-2020-12762
- CVE-2020-13435
- CVE-2020-14145
- CVE-2020-14155
- CVE-2020-16135
- CVE-2020-17541
- CVE-2020-24370
- CVE-2020-35521
- CVE-2020-35522
- CVE-2020-35523
- CVE-2020-35524
- CVE-2020-36330
- CVE-2020-36331
- CVE-2020-36332
- CVE-2021-3200

- CVE-2021-3426
- CVE-2021-3445
- CVE-2021-3481
- CVE-2021-3572
- CVE-2021-3580
- CVE-2021-3712
- CVE-2021-3800
- CVE-2021-20231
- CVE-2021-20232
- CVE-2021-20266
- CVE-2021-20317
- CVE-2021-21409
- CVE-2021-22876
- CVE-2021-22898
- CVE-2021-22925
- CVE-2021-27645
- CVE-2021-28153
- CVE-2021-31535
- CVE-2021-33560
- CVE-2021-33574
- CVE-2021-35942
- CVE-2021-36084
- CVE-2021-36085
- CVE-2021-36086
- CVE-2021-36087
- CVE-2021-37136
- CVE-2021-37137
- CVE-2021-42574
- CVE-2021-43267

- CVE-2021-43527
- CVE-2021-44228
- CVE-2021-45046

1.34. OPENSHIFT LOGGING 5.2.3

이번 릴리스에는 RHSA-2021:4032 OpenShift Logging 버그 수정 릴리스 5.2.3이 포함되어 있습니다.

1.34.1. 버그 수정

- 이번 업데이트 이전에는 일부 경고에 네임스페이스 라벨이 포함되지 않았습니다. 이 누락은 OpenShift Container Platform에서 경고 규칙을 작성하기 위한 OpenShift 모니터링 팀의 지침을 준수하지 않습니다. 이번 업데이트를 통해 Elasticsearch Operator의 모든 경고에는 네임스페이스 레이블이 포함되어 있으며 OpenShift Container Platform에서 경고 규칙을 작성하기 위한 모든 지침을 따르십시오. (LOG-1857)
- 이번 업데이트 이전에는 이전 릴리스에서 의도적으로 비활성화된 JSON 메시지 구문 분석이 도입된 회귀 문제가 있었습니다. 이번 업데이트에서는 JSON 구문 분석을 다시 활성화합니다. 또한 구문 분석된 JSON 메시지의 수준 필드를 기반으로 로그 항목 수준을 설정하거나 regex를 사용하여 메시지 필드에서 일치를 추출합니다. (LOG-1759)

1.34.2. CVE

예 1.24. CVE를 확장하려면 클릭합니다.

- CVE-2021-23369
  - BZ-1948761
- CVE-2021-23383
  - BZ-1956688
- CVE-2018-20673
- CVE-2019-5827

- CVE-2019-13750
- CVE-2019-13751
- CVE-2019-17594
- CVE-2019-17595
- CVE-2019-18218
- CVE-2019-19603
- CVE-2019-20838
- CVE-2020-12762
- CVE-2020-13435
- CVE-2020-14155
- CVE-2020-16135
- CVE-2020-24370
- CVE-2021-3200
- CVE-2021-3426

- CVE-2021-3445
- CVE-2021-3572
- CVE-2021-3580
- CVE-2021-3778
- CVE-2021-3796
- CVE-2021-3800
- CVE-2021-20231
- CVE-2021-20232
- CVE-2021-20266
- CVE-2021-22876
- CVE-2021-22898
- CVE-2021-22925
- CVE-2021-23840
- CVE-2021-23841

- CVE-2021-27645
- CVE-2021-28153
- CVE-2021-33560
- CVE-2021-33574
- CVE-2021-35942
- CVE-2021-36084
- CVE-2021-36085
- CVE-2021-36086
- CVE-2021-36087

1.35. OPENSHIFT LOGGING 5.2.2

이번 릴리스에는 RHBA-2021:3747 OpenShift Logging 버그 수정 릴리스 5.2.2가 포함되어 있습니다.

1.35.1. 버그 수정

- 이번 업데이트 이전에는 필요한 버퍼 공간을 사용할 수 없는 경우에도 **ClusterLogging** 사용자 정의 리소스(CR)에서 **totalLimitSize** 필드의 값을 **Fluentd total_limit_size** 필드에 적용했습니다. 이번 업데이트를 통해 **CR**은 두 개의 **totalLimitSize** 또는 **'default'** 값의 **lesser**를 **Fluentd total_limit_size** 필드에 적용하여 문제를 해결합니다. (LOG-1738)

- 이번 업데이트 이전에는 이전 릴리스 구성에 발생한 회귀 문제로 인해 수집기가 종료 전에 버퍼링된 메시지를 플러시하여 종료에 대한 지연을 생성하고 수집기 **Pod**를 다시 시작합니다. 이번

업데이트를 통해 **Fluentd**는 더 이상 종료 시 버퍼를 플러시하지 않아 문제를 해결합니다. (LOG-1739)

- 이번 업데이트 이전에는 번들 매니페스트의 문제로 **OpenShift Container Platform 4.9**의 **OLM**을 통해 **Elasticsearch Operator**를 설치할 수 없었습니다. 이번 업데이트를 통해 매니페스트를 번들로 다시 번들로 **4.9.**(LOG-1780)에서 설치를 다시 활성화합니다.

1.35.2. CVE

예 1.25. CVE를 확장하려면 클릭합니다.

- CVE-2020-25648
- CVE-2021-22922
- CVE-2021-22923
- CVE-2021-22924
- CVE-2021-36222
- CVE-2021-37576
- CVE-2021-37750
- CVE-2021-38201

1.36. OPENSHIFT LOGGING 5.2.1

이번 릴리스에는 RHBA-2021:3550 OpenShift Logging 버그 수정 릴리스 5.2.1이 포함되어 있습니다.

1.36.1. 버그 수정

- 이번 업데이트 이전에는 릴리스 파이프라인 스크립트의 문제로 인해 현재 릴리스 번호를 반영하는 대신 **olm.skipRange** 필드의 값이 **5.2.0** 에서 변경되지 않았습니다. 이번 업데이트에서는 릴리스 번호가 변경될 때 이 필드의 값을 업데이트하기 위해 파이프라인 스크립트가 수정되었습니다. (LOG-1743)

1.36.2. CVE

(없음)

1.37. OPENSHIFT LOGGING 5.2.0

이번 릴리스에는 RHBA-2021:3393 OpenShift Logging 버그 수정 릴리스 5.2.0이 포함되어 있습니다.

1.37.1. 새로운 기능 및 개선 사항

- 이번 업데이트를 통해 애플리케이션 및 인프라 모니터링을 제공하는 **Amazon CloudMonitor**에 로그 데이터를 전달할 수 있습니다. 자세한 내용은 Amazon CloudView로 로그 전달을 참조하십시오. (LOG-1173)

- 이번 업데이트를 통해 수평으로 확장 가능한 다중 테넌트 로그 집계 시스템인 **Loki**로 로그 데이터를 전달할 수 있습니다. 자세한 내용은 Loki로 로그 전달을 참조하십시오. (LOG-684)

- 이번 업데이트에서 **Fluentd** 전달 프로토콜을 사용하여 **TLS** 암호화 연결을 통해 로그 데이터를 전달하는 경우 이제 암호화된 개인 키 파일을 사용하고 클러스터 로그 전달자 구성에서 암호를 지정할 수 있습니다. 자세한 내용은 Fluentd 전달 프로토콜을 사용하여 로그 전달을 참조하십시오. (LOG-1525)

- 이러한 개선된 기능을 통해 사용자 이름과 암호를 사용하여 외부 **Elasticsearch** 인스턴스에 대한 로그 전달 연결을 인증할 수 있습니다. 예를 들어 타사가 **Elasticsearch** 인스턴스를 작동하기 때문에 상호 **TLS(mTLS)**를 사용할 수 없는 경우 **HTTP** 또는 **HTTPS**를 사용하고 사용자 이름과 암호가 포함된 시크릿을 설정할 수 있습니다. 자세한 내용은 외부 Elasticsearch 인스턴스로 로그 전달을 참조하십시오. (LOG-1022)

- 이번 업데이트를 통해 로깅 서버로 전달하기 위해 **OVN** 네트워크 정책 감사 로그를 수집할 수 있습니다. (LOG-1526)

- 기본적으로 **OpenShift Container Platform 4.5**에 도입된 데이터 모델은 다른 네임스페이스의 로그에 단일 인덱스를 제공했습니다. 이 변경으로 인해 가장 많은 로그를 생성한 네임스페이스

를 확인하기 어려웠습니다.

현재 릴리스에서는 **OpenShift Container Platform** 콘솔의 로깅 대시보드에 네임스페이스 메트릭이 추가되어 있습니다. 이러한 메트릭을 사용하면 지정된 타임 스탬프에 대해 로그를 생성하는 네임스페이스와 각 네임스페이스에서 생성하는 로그 수를 확인할 수 있습니다.

이러한 메트릭을 보려면 **OpenShift Container Platform** 웹 콘솔의 관리자 관점을 열고 **Observe → Dashboards → Logging/Elasticsearch** 로 이동합니다. (LOG-1680)

- 현재 릴리스인 **OpenShift Logging 5.2**는 다음 두 가지 새 지표를 활성화합니다. 지정된 타임 스탬프 또는 기간의 경우 개별 컨테이너에서 생성하거나 기록한 총 로그와 수집기에서 수집한 총 로그를 확인할 수 있습니다. 이러한 메트릭은 네임스페이스, **Pod** 및 컨테이너 이름으로 레이블이 지정되어 각 네임스페이스 및 **Pod**가 수집 및 생성하는 로그를 확인할 수 있습니다. (LOG-1213)

1.37.2. 버그 수정

- 이번 업데이트 이전에는 **OpenShift Elasticsearch Operator**가 인덱스 관리 **cronjob**을 생성할 때 **POLICY_MAPPING** 환경 변수를 두 번 추가되어 **apiserver**에서 중복 보고했습니다. 이번 업데이트에서는 **POLICY_MAPPING** 환경 변수가 **cronjob**당 한 번만 설정되고 **apiserver**에서 중복 보고가 없도록 문제가 해결되었습니다. (LOG-1130)

- 이번 업데이트 이전에는 **Elasticsearch** 클러스터를 **0** 노드로 일시 중단해도 인덱스 관리 **cronjob**이 일시 중단되지 않아 이러한 **cronjob**이 최대 백오프 상태가 되었습니다. 그런 다음 **Elasticsearch** 클러스터를 일시 중지하지 않은 후 최대 백오프 도달으로 인해 이러한 **cronjob**이 중지되었습니다. 이번 업데이트에서는 **cronjob** 및 클러스터를 일시 중단하여 문제가 해결되었습니다. (LOG-1268)

- 이번 업데이트 이전에는 **OpenShift Container Platform** 콘솔의 로깅 대시보드에서 상위 **10** 개 로그 생성 컨테이너 목록에 "차트 네임스페이스" 레이블이 누락되어 잘못된 메트릭 이름 **fluentd_input_status_total_bytes_logged**가 지정되었습니다. 이번 업데이트를 통해 차트에는 네임스페이스 레이블과 올바른 메트릭 이름, **log_logged_bytes_total**이 표시됩니다. (LOG-1271)

- 이번 업데이트 이전에는 인덱스 관리 **cronjob**이 오류와 함께 종료되면 오류 종료 코드를 보고하지 않고 대신 해당 작업 상태가 "완료"로 표시되었습니다. 이번 업데이트에서는 오류로 종료되는 인덱스 관리 **cronjob**의 오류 종료 코드를 보고하여 문제가 해결되었습니다. (LOG-1273)

- **priorityclasses.v1beta1.scheduling.k8s.io**가 **1.22**에서 제거되고 **priorityclasses.v1.scheduling.k8s.io** (**v1beta1**이 **v1**로 대체됨)로 대체되었습니다. 이번 업데이트 이전에는 **v1beta1**이 여전히 존재했기 때문에 **priorityclasses**에 대한

APIRemovedInNextReleaseInUse 경고가 생성되었습니다. 이번 업데이트에서는 v1beta1을 v1로 교체하여 문제가 해결되었습니다. 경고는 더 이상 생성되지 않습니다. (LOG-1385)

- 이전에는 OpenShift Elasticsearch Operator 및 Red Hat OpenShift Logging Operator에 연결이 끊긴 환경에서 실행할 수 있는 OpenShift Container Platform 웹 콘솔 목록에 표시되는 데 필요한 주석이 없었습니다. 이번 업데이트에서는 operators.openshift.io/infrastructure-features: '["Disconnected"]' 주석을 추가하여 연결이 끊긴 환경에서 실행되는 Operator 목록에 표시됩니다. (LOG-1420)

- 이번 업데이트 이전에 Red Hat OpenShift Logging Operator Pod는 성능 최적화된 단일 노드 클러스터의 고객 워크로드용으로 예약된 CPU 코어에서 예약되었습니다. 이번 업데이트를 통해 클러스터 로깅 Operator Pod가 올바른 CPU 코어에 예약됩니다. (LOG-1440)

- 이번 업데이트 이전에는 일부 로그 항목에 인식할 수 없는 UTF-8 바이트가 있어 Elasticsearch가 메시지를 거부하고 전체 버퍼링된 페이로드를 차단했습니다. 이번 업데이트를 통해 거부된 페이로드가 잘못된 로그 항목을 삭제하고 나머지 항목을 다시 제출하여 문제가 해결되었습니다. (LOG-1499)

- 이번 업데이트 이전에는 kibana-proxy Pod가 CrashLoopBackoff 상태에 들어가서 Invalid 설정을 기록했습니다. cookie_secret은 pass_access_token == true 또는 cookie_refresh != 0 일 때 AES 암호를 생성하려면 32바이트여야 합니다. 정확한 실제 바이트 수는 다를 수 있습니다. 이번 업데이트를 통해 Kibana 세션 시크릿 생성이 수정되었으며 이 오류로 인해 kibana-proxy Pod가 더 이상 CrashLoopBackoff 상태에 들어가지 않습니다. (LOG-1446)

- 이번 업데이트 이전에는 AWS CloudMonitor Fluentd 플러그인이 모든 로그 수준에서 Fluentd 로그에 AWS API 호출을 기록하여 추가 OpenShift Container Platform 노드 리소스를 사용합니다. 이번 업데이트를 통해 AWS CloudMonitor Fluentd 플러그인은 "디버그" 및 "추적" 로그 수준에서만 AWS API 호출을 기록합니다. 이렇게 하면 기본 "경고" 로그 수준에서 Fluentd는 추가 노드 리소스를 사용하지 않습니다. (LOG-1071)

- 이번 업데이트 이전에는 Elasticsearch OpenDistro 보안 플러그인으로 인해 사용자 인덱스 마이그레이션이 실패했습니다. 이번 업데이트에서는 최신 버전의 플러그인을 제공하여 문제를 해결했습니다. 이제 인덱스 마이그레이션이 오류 없이 진행됩니다. (LOG-1276)

- 이번 업데이트 이전에는 OpenShift Container Platform 콘솔의 로깅 대시보드에서 상위 10개의 로그 생성 컨테이너 목록에 데이터 포인트가 없었습니다. 이번 업데이트에서는 문제가 해결되어 대시보드에 모든 데이터 포인트가 표시됩니다. (LOG-1353)

- 이번 업데이트 이전에는 chunkLimitSize 및 totalLimitSize 값을 조정하여 Fluentd 로그 전달자의 성능을 튜닝하는 경우 값이 너무 낮다는 Setting queued_chunks_limit_size for each

buffer to 메시지가 보고되었습니다. 현재 업데이트를 통해 이 메시지가 올바른 값을 보고하도록 이 문제가 해결되었습니다. (LOG-1411)

- 이번 업데이트 이전에는 Kibana OpenDistro 보안 플러그인으로 인해 사용자 인덱스 마이그레이션이 실패했습니다. 이번 업데이트에서는 최신 버전의 플러그인을 제공하여 문제를 해결했습니다. 이제 인덱스 마이그레이션이 오류 없이 진행됩니다. (LOG-1558)

- 이번 업데이트 이전에는 네임스페이스 입력 필터를 사용하면 해당 네임스페이스에 있는 로그가 다른 입력에 표시되지 않았습니다. 이번 업데이트를 통해 로그를 수락할 수 있는 모든 입력으로 전송됩니다. (LOG-1570)

- 이번 업데이트 이전에는 viaq/logerr 종속성에 대한 라이센스 파일이 누락되어 라이센스 스캐너가 성공하지 못하고 중단되었습니다. 이번 업데이트를 통해 viaq/logerr 종속성은 Apache 2.0에 따라 라이센스가 부여되고 라이센스 스캐너가 성공적으로 실행됩니다. (LOG-1590)

- 이번 업데이트 이전에는 elasticsearch-operator-bundle 빌드 파이프라인에서 curator5에 대한 올바르지 않은 태그로 인해 dummy SHA1에 고정된 이미지 가져오기가 발생했습니다. 이번 업데이트를 통해 빌드 파이프라인은 curator5의 logging-curator5-rhel8 참조를 사용하여 인덱스 관리 cronjob이 registry.redhat.io에서 올바른 이미지를 가져올 수 있습니다. (LOG-1624)

- 이번 업데이트 이전에는 ServiceAccount 권한 문제로 인해 no permissions for [indices:admin/aliases/get]과 같은 오류가 발생했습니다. 이번 업데이트를 통해 권한 수정으로 문제가 해결되었습니다. (LOG-1657)

- 이번 업데이트 이전에는 Red Hat OpenShift Logging Operator의 CRD(Custom Resource Definition)에 Loki 출력 유형이 누락되어 승인 컨트롤러에서 ClusterLogForwarder 사용자 정의 리소스 오브젝트를 거부했습니다. 이번 업데이트를 통해 CRD에 Loki가 출력 유형으로 포함되어 관리자가 Loki 서버에 로그를 보내도록 ClusterLogForwarder를 구성할 수 있습니다. (LOG-1683)

- 이번 업데이트 이전에는 ServiceAccounts의 OpenShift Elasticsearch Operator 조정이 보안이 포함된 타사 소유 필드를 덮어썼습니다. 이 문제로 인해 시크릿을 자주 재생성하여 메모리와 CPU가 급증했습니다. 이번 업데이트에서는 이러한 문제가 해결되었습니다. 이제 OpenShift Elasticsearch Operator가 타사 소유 필드를 덮어쓰지 않습니다. (LOG-1714)

- 이번 업데이트 이전에는 ClusterLogging CR(사용자 정의 리소스) 정의에서 flush_interval 값을 지정했지만 flush_mode를 interval로 설정하지 않은 경우 Red Hat OpenShift Logging Operator에서 Fluentd 구성을 생성했습니다. 그러나 Fluentd 수집기에서 런타임 시 오류가 발생했습니다. 이번 업데이트를 통해 Red Hat OpenShift Logging Operator는 ClusterLogging CR

정의의 유효성을 검사하고 두 필드가 모두 지정된 경우에만 **Fluentd** 구성을 생성합니다. (LOG-1723)

1.37.3. 확인된 문제

- 외부 **Elasticsearch** 서버로 로그를 전달한 다음 사용자 이름 및 암호와 같은 파이프라인 시크릿에 구성된 값을 변경하는 경우 **Fluentd** 전달자가 새 시크릿을 로드하지만 이전 값을 사용하여 외부 **Elasticsearch** 서버에 연결합니다. 이 문제는 **Red Hat OpenShift Logging Operator**가 현재 콘텐츠 변경 사항에 대한 시크릿을 모니터링하지 않기 때문에 발생합니다. (LOG-1652)

  문제를 해결하기 위해 시크릿을 변경하는 경우 다음을 입력하여 **Fluentd Pod**를 강제로 재배포할 수 있습니다.

  ```
  $ oc delete pod -l component=collector
  ```

1.37.4. 사용되지 않거나 삭제된 기능

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

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

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

OpenShift Container Platform 4.6에서 현재까지 다음 레거시 방법을 사용하여 로그를 전달하는 것은 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다.

- 레거시 **Fluentd** 방법을 사용하여 로그 전달
- 레거시 **syslog** 방법을 사용하여 로그 전달

대신 다음과 같은 비레거시 방법을 사용합니다.

- **Fluentd fohttps://www.redhat.com/security/data/cve/CVE-2021-22922.htmlrward** 프로토콜을 사용하여 로그 전달

- syslog 프로토콜을 사용하여 로그 전달

1.37.6. CVE

예 1.26. CVE를 확장하려면 클릭합니다.

- CVE-2021-22922
- CVE-2021-22923
- CVE-2021-22924
- CVE-2021-32740
- CVE-2021-36222
- CVE-2021-37750

2장. RED HAT OPENSHIFT LOGGING 이해

클러스터 관리자는 **OpenShift Logging**을 배포하여 **OpenShift Container Platform** 클러스터의 모든 로그(예: 노드 시스템 감사 로그, 애플리케이션 컨테이너 로그 및 인프라 로그)를 집계할 수 있습니다. **OpenShift Logging**은 클러스터 전체에서 이러한 로그를 집계하여 기본 로그 저장소에 저장합니다. Kibana 웹 콘솔을 사용하여 로그 데이터를 시각화할 수 있습니다.

OpenShift Logging에서는 다음 유형의 로그를 집계합니다.

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

참고

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

2.1. OPENSHIFT CONTAINER PLATFORM LOGGING에 대한 일반 용어집

이 용어집은 **OpenShift Container Platform** 로깅 콘텐츠에 사용되는 일반적인 용어를 정의합니다.

주석

주석을 사용하여 메타데이터를 오브젝트에 연결할 수 있습니다.

CLO(Cluster Logging Operator)

Cluster Logging Operator는 애플리케이션, 인프라 및 감사 로그의 수집 및 전달을 제어하는 **API** 세트를 제공합니다.

CR(사용자 정의 리소스)

CR은 Kubernetes API의 확장입니다. OpenShift Container Platform 로깅 및 로그 전달을 구성하려면 ClusterLogging 및 ClusterLogForwarder 사용자 정의 리소스를 사용자 정의할 수 있습니다.

이벤트 라우터

이벤트 라우터는 OpenShift Container Platform 이벤트를 감시하는 Pod입니다. OpenShift Container Platform 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 Platform에서는 ELasticsearch를 OpenShift Container Platform 로깅의 기본 로그 저장소로 사용합니다.

Elasticsearch Operator

Elasticsearch Operator는 OpenShift Container Platform에서 Elasticsearch 클러스터를 실행하는 데 사용됩니다. Elasticsearch Operator는 Elasticsearch 클러스터 작업에 대한 셀프 서비스를 제공하며 OpenShift Container Platform 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

OpenShift Container Platform Logging Log Forwarding API를 사용하면 JSON 로그를 구조화된 오브젝트로 구문 분석하고 OpenShift Container Platform Logging 관리 Elasticsearch 또는 Log Forwarding API에서 지원하는 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 **Pod**와 같은 오브젝트 서브 세트를 구성하고 선택하는 데 사용할 수 있는 키-값 쌍입니다.

로깅

OpenShift Container Platform Logging을 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. 기본 **Elasticsearch** 로그 저장소를 사용하거나 외부 로그 저장소로 로그를 전달할 수 있습니다. 기본 로그 저장소는 테스트를 거쳐 단기 스토리지용으로 최적화되었습니다.

로그 시각화 프로그램

로그 시각화 프로그램은 로그, 그래프, 차트 및 기타 지표와 같은 정보를 보는 데 사용할 수 있는 **UI**(사용자 인터페이스) 구성 요소입니다. 최신 구현은 **Kibana**입니다.

node

노드는 **OpenShift Container Platform** 클러스터의 작업자 시스템입니다. 노드는 **VM**(가상 머신) 또는 물리적 머신입니다.

Operator

Operator는 **OpenShift Container Platform** 클러스터에서 **Kubernetes** 애플리케이션을 패키징, 배포 및 관리하는 기본 방법입니다. **Operator**는 사람의 운영 지식을 패키지하고 고객과 공유하는 소프트웨어로 인코딩합니다.

Pod

Pod는 **Kubernetes**에서 가장 작은 논리 단위입니다. **Pod**는 하나 이상의 컨테이너로 구성되며 작업자 노드에서 실행됩니다.

RBAC(역할 기반 액세스 제어)

RBAC는 클러스터 사용자 및 워크로드가 역할을 실행하는 데 필요한 리소스에만 액세스할 수 있도록 하는 핵심 보안 제어입니다.

shard

Elasticsearch는 Fluentd의 로그 데이터를 데이터 저장소 또는 인덱스로 구성한 다음 각 인덱스를 shards라는 여러 조각으로 세분화합니다.

taint

테인트를 사용하면 Pod가 적절한 노드에 예약됩니다. 노드에 하나 이상의 테인트를 적용할 수 있습니다.

허용 오차

Pod에 허용 오차를 적용할 수 있습니다. 허용 오차를 사용하면 스케줄러에서 일치하는 테인트를 사용하여 Pod를 예약할 수 있습니다.

웹 콘솔

OpenShift Container Platform을 관리할 UI(사용자 인터페이스)입니다.

2.2. OPENSHIFT LOGGING 배포 이해

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

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

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

자세한 내용은 로그 수집기 구성을 참조하십시오.

2.2.1. JSON OpenShift Container Platform 로깅 정보

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

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

자세한 내용은 JSON 로깅 정보를 참조하십시오.

2.2.2. Kubernetes 이벤트 수집 및 저장 정보

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

자세한 내용은 Kubernetes 이벤트 수집 및 저장을 참조하십시오.

2.2.3. OpenShift Container Platform Logging 업데이트 정보

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

- Elasticsearch Operator
- Cluster Logging Operator

자세한 내용은 OpenShift Container Platform 로깅 업데이트 정보를 참조하십시오.

2.2.4. 클러스터 대시보드 보기 정보

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

자세한 내용은 클러스터 대시보드 보기 정보를 참조하십시오.

2.2.5. OpenShift Container Platform 로깅 문제 해결 정보

다음 작업을 수행하여 로깅 문제를 해결할 수 있습니다.

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

2.2.6. OpenShift Container Platform 로깅 설치 제거 정보

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

자세한 내용은 OpenShift Container Platform 로깅 설치 제거를 참조하십시오.

2.2.7. 필드 내보내기 정보

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

자세한 내용은 필드 내보내기 정보를 참조하십시오.

2.2.8. OpenShift Logging 구성 요소 정보

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

OpenShift Logging의 주요 구성 요소는 다음과 같습니다.

- 수집 - 클러스터에서 로그를 수집하고 형식을 지정한 후 로그 저장소로 전달하는 구성 요소입니다. 최신 구현은 Fluentd입니다.
- 로그 저장소 - 로그가 저장되는 위치입니다. 기본 구현은 Elasticsearch입니다. 기본 Elasticsearch 로그 저장소를 사용하거나 외부 로그 저장소로 로그를 전달할 수 있습니다. 기본 로그 저장소는 테스트를 거쳐 단기 스토리지용으로 최적화되었습니다.
- 시각화 - 로그, 그래프, 차트 등을 보는 데 사용할 수 있는 UI 구성 요소입니다. 최신 구현은 Kibana입니다.

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

2.2.9. 로깅 수집기 정보

OpenShift Container Platform은 Fluentd를 사용하여 컨테이너 및 노드 로그를 수집합니다.

기본적으로 로그 수집기는 다음 소스를 사용합니다.

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

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

로깅 수집기는 데몬 세트로 각 OpenShift Container Platform 노드에 Pod를 배포합니다. 시스템 및

인프라 로그는 journald가 운영 체제, 컨테이너 런타임 및 OpenShift Container Platform의 로그 메시지를 사용하여 생성합니다. 애플리케이션 로그는 CRI-O 컨테이너 엔진에 의해 생성됩니다. Fluentd는 이러한 소스에서 로그를 수집하여 OpenShift Container Platform의 구성에 따라 내부 또는 외부로 전달합니다.

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

중요

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

자세한 내용은 로그 수집기 구성을 참조하십시오.

2.2.10. 로그 저장소 정보

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

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

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

참고

인덱스 템플릿의 기본 shard 수는 Elasticsearch 데이터 노드 수와 같습니다.

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

참고

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

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

자세한 내용은 로그 저장소 구성을 참조하십시오.

2.2.11. 로깅 시각화 정보

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

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

자세한 내용은 로그 시각화 프로그램 구성을 참조하십시오.

2.2.12. 이벤트 라우팅 정보

이벤트 라우터는 OpenShift Logging으로 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.

2.2.13. 로그 전송 정보

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

자세한 내용은 타사 시스템으로 로그 전달을 참조하십시오.

3장. OPENSHIFT LOGGING 설치

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

OpenShift Container Platform에 OpenShift Logging을 배포하는 프로세스에는 다음이 포함됩니다.

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

3.1. 웹 콘솔을 사용하여 OPENSHIFT LOGGING 설치

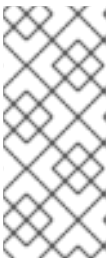
OpenShift Container Platform 웹 콘솔을 사용하여 OpenShift Elasticsearch Operator 및 Red Hat OpenShift Logging Operator를 설치할 수 있습니다.

참고

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

사전 요구 사항

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



참고

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

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

프로세스

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

1. OpenShift Elasticsearch Operator를 설치합니다.

   a. OpenShift Container Platform 웹 콘솔에서 Operator → OperatorHub를 클릭합니다.

   b. 사용 가능한 Operator 목록에서 OpenShift Elasticsearch Operator를 선택한 다음 설치를 클릭합니다.

   c. 설치 모드에서 클러스터의 모든 네임스페이스가 선택되어 있는지 확인합니다.

   d. 설치된 네임스페이스에서 openshift-operators-redhat이 선택되어 있는지 확인합니다.

      openshift-operators-redhat 네임스페이스를 지정해야 합니다. openshift-operators 네임스페이스에 신뢰할 수 없는 Community Operator가 포함될 수 있고, 여기에서 OpenShift Container Platform 지표와 동일한 이름의 지표를 게시하면 충돌이 발생합니다.

   e. 이 네임스페이스에서 Operator 권장 클러스터 모니터링 사용을 선택합니다.

      이 옵션은 네임스페이스 오브젝트에서 openshift.io/cluster-monitoring: "true" 레이블을 설정합니다. 클러스터 모니터링이 openshift-operators-redhat 네임스페이스를 스크랩하도록 하려면 이 옵션을 선택해야 합니다.

   f. stable-5.x을 업데이트 채널로 선택합니다.

   g. 승인 전략을 선택합니다.

- 자동 전략을 사용하면 **Operator** 새 버전이 준비될 때 **OLM(Operator Lifecycle Manager)**이 자동으로 **Operator**를 업데이트할 수 있습니다.

- 수동 전략을 사용하려면 적절한 자격 증명을 가진 사용자가 **Operator** 업데이트를 승인해야 합니다.

h. 설치를 클릭합니다.

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

j. 상태가 성공인 모든 프로젝트에 **OpenShift Elasticsearch Operator**가 나열되어 있는지 확인합니다.

2. **Red Hat OpenShift Logging Operator**를 설치합니다.

a. **OpenShift Container Platform** 웹 콘솔에서 **Operator** → **OperatorHub**를 클릭합니다.

b. 사용 가능한 **Operator** 목록에서 **Red Hat OpenShift Logging**을 선택한 다음 설치를 클릭합니다.

c. 설치 모드에서 클러스터의 특정 네임스페이스가 선택되어 있는지 확인합니다.

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

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

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

f. **stable-5.x**을 업데이트 채널로 선택합니다.

g. 승인 전략을 선택합니다.

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

h. 설치를 클릭합니다.

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

j. **Red Hat OpenShift Logging**이 **openshift-logging** 프로젝트에 성공 상태로 나열되어 있는지 확인합니다.

Operator가 설치된 것으로 나타나지 않으면 다음과 같이 추가 문제 해결을 수행합니다.

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

3. **OpenShift Logging** 인스턴스를 생성합니다.

a. 관리 → 사용자 정의 리소스 정의 페이지로 전환합니다.

b. 사용자 정의 리소스 정의 페이지에서 **ClusterLogging**을 클릭합니다.

c. 사용자 정의 리소스 정의 상세 정보 페이지의 작업 메뉴에서 인스턴스 보기를 선택합니다.

d. **ClusterLoggings** 페이지에서 **ClusterLogging** 생성을 클릭합니다.

데이터를 로드하기 위해 페이지를 새로 고쳐야 할 수도 있습니다.

e. **YAML** 필드에서 코드를 다음으로 교체합니다.

참고

이 기본 **OpenShift Logging** 구성은 다양한 환경을 지원해야 합니다. **OpenShift Logging** 클러스터에 수행할 수 있는 수정 사항에 대한 정보는 **OpenShift Logging** 구성 요소 튜닝 및 구성 주제를 검토하십시오.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance" 1
  namespace: "openshift-logging"
spec:
  managementState: "Managed" 2
  logStore:
    type: "elasticsearch" 3
    retentionPolicy: 4
      application:
        maxAge: 1d
      infra:
        maxAge: 7d
      audit:
        maxAge: 7d
    elasticsearch:
      nodeCount: 3 5
      storage:
        storageClassName: "<storage_class_name>" 6
        size: 200G
      resources: 7
          limits:
            memory: "16Gi"
          requests:
            memory: "16Gi"
      proxy: 8
        resources:
          limits:
            memory: 256Mi
```

```
      requests:
        memory: 256Mi
    redundancyPolicy: "SingleRedundancy"
visualization:
  type: "kibana"  9
  kibana:
    replicas: 1
collection:
  logs:
    type: "fluentd"  10
    fluentd: {}
```

1

이름은 **instance**이어야 합니다.

2

OpenShift Logging 관리 상태입니다. 경우에 따라 **OpenShift Logging** 기본값을 변경하는 경우 이를 **Unmanaged**로 설정해야 합니다. 그러나 관리되지 않는 배포는 **OpenShift Logging**이 다시 **Managed** 상태로 될 때까지 업데이트를 받지 않습니다.

3

Elasticsearch 구성을 위한 설정입니다. **CR**을 사용하여 **shard** 복제 정책 및 영구 스토리지를 구성할 수 있습니다.

4

Elasticsearch가 각 로그 소스를 유지해야 하는 시간을 지정합니다. 정수 및 시간 지정을 입력합니다(주(w), 시간(h/H), 분(m) 및 초(s)). 예를 들어 **7**일은 **7d**입니다. **maxAge**보다 오래된 로그는 삭제됩니다. 각 로그 소스에 대한 보존 정책을 지정해야 합니다. 그렇지 않으면 해당 소스에 대해 **Elasticsearch** 인덱스가 생성되지 않습니다.

5

Elasticsearch 노드 수를 지정합니다. 이 목록 뒤에 나오는 참고 사항을 참조하십시오.

6

Elasticsearch 스토리지의 기존 스토리지 클래스 이름을 입력합니다. 최상의 성능을 위해서는 블록 스토리지를 할당하는 스토리지 클래스를 지정합니다. 스토리지 클래스를 지정하지 않으면 **OpenShift 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 **16Gi**이고 **CPU** 요청 시 **1**입니다.

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 **256Mi**이고 **CPU** 요청 시 **100m**입니다.

9

Kibana 구성을 위한 설정입니다. **CR**을 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. **CR**을 사용하여 **Fluentd CPU** 및 메모리 제한을 구성할 수 있습니다. 자세한 내용은 **Fluentd** 구성을 참조하십시오.

참고

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

예를 들어 nodeCount = 4인 경우 다음 노드가 생성됩니다.

```
$ oc get deployment
```

출력 예

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

인덱스 템플릿의 기본 shard 수는 Elasticsearch 데이터 노드 수와 같습니다.

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

4. 설치를 확인합니다.

a. 워크로드 → Pod 페이지로 전환합니다.

b. **openshift-logging** 프로젝트를 선택합니다.

다음 목록과 유사한 **OpenShift Logging, Elasticsearch, Fluentd** 및 **Kibana**에 대한 여러 **Pod**가 표시됩니다.

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

추가 리소스

- **OperatorHub**에서 **Operator** 설치

3.2. 설치 후 작업

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

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

3.3. CLI를 사용하여 OPENSHIFT LOGGING 설치

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

사전 요구 사항

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

참고

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

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

프로세스

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

1. OpenShift Elasticsearch Operator의 네임스페이스를 생성합니다.

   a. OpenShift Elasticsearch Operator를 위한 네임스페이스 오브젝트 YAML 파일(예: eo-

namespace.yaml)을 생성합니다.

```
apiVersion: v1
kind: Namespace
metadata:
 name: openshift-operators-redhat (1)
 annotations:
  openshift.io/node-selector: ""
 labels:
  openshift.io/cluster-monitoring: "true" (2)
```

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

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

b. 네임스페이스를 생성합니다.

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

예를 들면 다음과 같습니다.

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

2. Red Hat OpenShift Logging Operator의 네임스페이스를 생성합니다.

a. Red Hat OpenShift Logging Operator를 위한 네임스페이스 오브젝트 YAML 파일(예: olo-namespace.yaml)을 생성합니다.

```
apiVersion: v1
kind: Namespace
metadata:
 name: openshift-logging
 annotations:
```

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

b. 네임스페이스를 생성합니다.

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

예를 들면 다음과 같습니다.

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

3. 다음 오브젝트를 생성하여 **OpenShift Elasticsearch Operator**를 설치합니다.

a. **OpenShift Elasticsearch Operator**를 위한 **Operator** 그룹 오브젝트 **YAML** 파일(예: **eo-og.yaml**)을 생성합니다.

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: openshift-operators-redhat
  namespace: openshift-operators-redhat (1)
spec: {}
```

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

b. **Operator** 그룹 오브젝트를 생성합니다.

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

예를 들면 다음과 같습니다.

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

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

서브스크립션의 예

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: "elasticsearch-operator"
  namespace: "openshift-operators-redhat" 1
spec:
  channel: "stable-5.1" 2
  installPlanApproval: "Automatic" 3
  source: "redhat-operators" 4
  sourceNamespace: "openshift-marketplace"
  name: "elasticsearch-operator"
```

1

openshift-operators-redhat 네임스페이스를 지정해야 합니다.

2

5.0, stable 또는 stable-5.<x>를 채널로 지정합니다. 다음 참고 사항을 참조하십시오.

3

자동 을 사용하면 새 버전이 사용 가능할 때 OLM(Operator Lifecycle Manager)이 자동으로 Operator를 업데이트할 수 있습니다. Operator 업데이트를 승인하려면 적절한 인증 정보를 가진 사용자가 수동 이 필요합니다.

4

redhat-operators를 지정합니다. OpenShift Container Platform 클러스터가 제한된 네트워크(연결이 끊긴 클러스터)에 설치된 경우 OLM(Operator Lifecycle Manager)을 구성할 때 생성된 CatalogSource 오브젝트의 이름을 지정합니다.

참고

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

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

d. 서브스크립션 오브젝트를 생성합니다.

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

예를 들면 다음과 같습니다.

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

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

e. **Operator** 설치를 확인합니다.

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

출력 예

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

```
Succeeded
openshift-apiserver-operator                elasticsearch-operator.5.1.0-
202007012112.p0    OpenShift Elasticsearch Operator   5.1.0-202007012112.p0
Succeeded
openshift-apiserver                         elasticsearch-operator.5.1.0-
202007012112.p0    OpenShift Elasticsearch Operator   5.1.0-202007012112.p0
Succeeded
openshift-authentication-operator           elasticsearch-operator.5.1.0-
202007012112.p0    OpenShift Elasticsearch Operator   5.1.0-202007012112.p0
Succeeded
openshift-authentication                    elasticsearch-operator.5.1.0-
202007012112.p0    OpenShift Elasticsearch Operator   5.1.0-202007012112.p0
Succeeded
...
```

각 네임스페이스에 **OpenShift Elasticsearch Operator**가 있어야 합니다. 버전 번호가 표시된 것과 다를 수 있습니다.

4. 다음 오브젝트를 생성하여 **Red Hat OpenShift Logging Operator**를 설치합니다.

a. **Red Hat OpenShift Logging Operator**를 위한 **OperatorGroup** 오브젝트 **YAML** 파일(예: **olo-og.yaml**)을 생성합니다.

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

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

b. **OperatorGroup** 개체를 생성합니다.

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

예를 들면 다음과 같습니다.

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

c.

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

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

(1)

openshift-logging 네임스페이스를 지정해야 합니다.

(2)

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

(3)

redhat-operators를 지정합니다. OpenShift Container Platform 클러스터가 제한된 네트워크(연결이 끊긴 클러스터)에 설치된 경우 OLM(Operator Lifecycle Manager)을 구성할 때 생성된 CatalogSource 오브젝트의 이름을 지정합니다.

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

예를 들면 다음과 같습니다.

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

Red Hat OpenShift Logging Operator는 openshift-logging 네임스페이스에 설치됩니다.

d.

Operator 설치를 확인합니다.

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

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

출력 예

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

5.

OpenShift Logging 인스턴스를 생성합니다.

a.

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

참고

이 기본 OpenShift Logging 구성은 다양한 환경을 지원해야 합니다. OpenShift Logging 클러스터에 수행할 수 있는 수정 사항에 대한 정보는 OpenShift Logging 구성 요소 튜닝 및 구성 주제를 검토하십시오.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance" 1
  namespace: "openshift-logging"
spec:
  managementState: "Managed"  2
  logStore:
    type: "elasticsearch"  3
    retentionPolicy: 4
      application:
```

```
      maxAge: 1d
    infra:
      maxAge: 7d
    audit:
      maxAge: 7d
  elasticsearch:
    nodeCount: 3 5
    storage:
      storageClassName: "<storage-class-name>" 6
      size: 200G
    resources: 7
      limits:
        memory: "16Gi"
      requests:
        memory: "16Gi"
    proxy: 8
      resources:
        limits:
          memory: 256Mi
        requests:
          memory: 256Mi
    redundancyPolicy: "SingleRedundancy"
visualization:
  type: "kibana" 9
  kibana:
    replicas: 1
collection:
  logs:
    type: "fluentd" 10
    fluentd: {}
```

1

이름은 **instance**이어야 합니다.

2

OpenShift Logging 관리 상태입니다. 경우에 따라 **OpenShift Logging** 기본값을 변경하는 경우 이를 **Unmanaged**로 설정해야 합니다. 그러나 관리되지 않는 배포는 **OpenShift Logging**이 다시 **Managed** 상태로 될 때까지 업데이트를 받지 않습니다. 배포를 다시 **Managed** 상태로 설정하면 수정한 내용이 취소될 수 있습니다.

3

Elasticsearch 구성을 위한 설정입니다. **CR**(사용자 정의 리소스)을 사용하여 **shard** 복제 정책 및 영구 스토리지를 구성할 수 있습니다.

4

Elasticsearch가 각 로그 소스를 유지해야 하는 시간을 지정합니다. 정수 및 시간 지정을 입력합니다(주(w), 시간(h/H), 분(m) 및 초(s)). 예를 들어 **7**일은 **7d**입니다. **maxAge**보다 오래된 로그는 삭제됩니다. 각 로그 소스에 대한 보존 정책을 지정해야 합

니다. 그렇지 않으면 해당 소스에 대해 Elasticsearch 인덱스가 생성되지 않습니다.

5

Elasticsearch 노드 수를 지정합니다. 이 목록 뒤에 나오는 참고 사항을 참조하십시오.

6

Elasticsearch 스토리지의 기존 스토리지 클래스 이름을 입력합니다. 최상의 성능을 위해서는 블록 스토리지를 할당하는 스토리지 클래스를 지정합니다. 스토리지 클래스를 지정하지 않으면 OpenShift Container Platform은 임시 스토리지로만 OpenShift Logging을 배포합니다.

7

필요에 따라 Elasticsearch에 대한 CPU 및 메모리 요청을 지정합니다. 이러한 값을 비워 두면 OpenShift Elasticsearch Operator는 대부분의 배포에 충분한 기본값을 설정합니다. 기본값은 메모리 요청 시 16Gi이고 CPU 요청 시 1입니다.

8

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

9

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

10

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

참고

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

예를 들어 nodeCount = 4인 경우 다음 노드가 생성됩니다.

```
$ oc get deployment
```

출력 예

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

인덱스 템플릿의 기본 shard 수는 Elasticsearch 데이터 노드 수와 같습니다.

b. 인스턴스를 생성합니다.

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

예를 들면 다음과 같습니다.

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

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

6. **openshift-logging** 프로젝트에 **Pod**를 나열하여 설치를 확인합니다.

다음 목록과 유사한 **OpenShift Logging, Elasticsearch, Fluentd** 및 **Kibana**에 대한 여러 **Pod**가 표시됩니다.

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

출력 예

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

3.4. 설치 후 작업

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

클러스터 네트워크 공급자가 네트워크 분리를 적용하는 경우 **OpenShift Logging Operator**가 포함된 프로젝트 간에 네트워크 트래픽을 허용합니다.

3.4.1. Kibana 인덱스 패턴 정의

인덱스 패턴은 시각화하려는 **Elasticsearch** 인덱스를 정의합니다. **Kibana**에서 데이터를 탐색하고 시각화하려면 인덱스 패턴을 생성해야 합니다.

사전 요구 사항

- **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 **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/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.

프로세스

Kibana에서 인덱스 패턴을 정의하고 시각화를 생성하려면 다음을 수행합니다.

1. **OpenShift Container Platform** 콘솔에서 **Application Launcher**

를 클릭하고 로깅을 선택합니다.

2. 관리 → 인덱스 패턴 → 인덱스 패턴 생성을 클릭하여 **Kibana** 인덱스 패턴을 생성합니다.

- 각 사용자는 프로젝트의 로그를 보려면 **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 로그인할 때 인덱스 패턴을 생성해야 합니다.

3. 새로운 인덱스 패턴에서 **Kibana** 시각화를 생성합니다.

3.4.2. 네트워크 분리가 활성화될 때 프로젝트 간 트래픽 허용

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

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

OpenShift Container Platform은 기본 **CNI(Container Network Interface)** 네트워크 공급자인 **OpenShift SDN**과 **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 필요

하지 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 만듭니다. 예를 들면 다음과 같습니다.

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

```
        network.openshift.io/policy-group: ingress
  podSelector: {}
  policyTypes:
  - Ingress
```

추가 리소스

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

4장. 로깅 배포 구성

4.1. 클러스터 로깅 사용자 정의 리소스 정보

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

4.1.1. 클러스터 로깅 사용자 정의 리소스 정보

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

CR을 작성하거나 수정하기 위한 지침이 이 문서에 적절하게 제공됩니다.

다음은 **OpenShift Logging**을 위한 일반적인 사용자 정의 리소스의 예입니다.

ClusterLogging 사용자 정의 리소스 (CR) 샘플

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance" 1
  namespace: "openshift-logging" 2
spec:
  managementState: "Managed" 3
  logStore:
    type: "elasticsearch" 4
    retentionPolicy:
      application:
        maxAge: 1d
      infra:
        maxAge: 7d
      audit:
        maxAge: 7d
    elasticsearch:
      nodeCount: 3
      resources:
        limits:
          memory: 16Gi
        requests:
          cpu: 500m
          memory: 16Gi
      storage:
        storageClassName: "gp2"
```

```
      size: "200G"
    redundancyPolicy: "SingleRedundancy"
visualization: 5
  type: "kibana"
  kibana:
    resources:
      limits:
        memory: 736Mi
      requests:
        cpu: 100m
        memory: 736Mi
    replicas: 1
collection: 6
  logs:
    type: "fluentd"
    fluentd:
      resources:
        limits:
          memory: 736Mi
        requests:
          cpu: 100m
          memory: 736Mi
```

1

CR 이름은 **instance**여야 합니다.

2

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

3

Red Hat OpenShift Logging Operator 관리 상태입니다. **Unmanaged**로 설정된 경우 Operator는 지원되지 않는 상태이며 업데이트되지 않습니다.

4

보존 정책, 노드 수, 리소스 요청 및 제한, 스토리지 클래스를 포함한 로그 저장소 설정

5

리소스 요청 및 제한, **Pod** 복제본 수를 포함한 시각화 프로그램 설정

6

리소스 요청 및 제한을 포함한 로그 수집기 설정

4.2. 로깅 수집기 구성

OpenShift Container Platform은 **Fluentd**를 사용하여 클러스터에서 작업 및 애플리케이션 로그를 수집하고 **Kubernetes Pod** 및 프로젝트 메타데이터로 데이터를 보강합니다.

로그 수집기의 **CPU** 및 메모리 제한을 구성하고 로그 수집기 **Pod**를 특정 노드로 이동할 수 있습니다. **ClusterLogging** 사용자 정의 리소스(**CR**)의 **spec.collection.log.fluentd** 스탠자를 통해 로그 수집기에 대해 지원되는 모든 수정을 수행할 수 있습니다.

4.2.1. 지원되지 않는 구성 정보

지원되는 **OpenShift Logging** 구성 방법은 이 설명서에 설명된 옵션을 사용하여 구성하는 것입니다. 다른 구성은 지원되지 않으므로 사용하지 마십시오. 구성 패러다임은 **OpenShift Container Platform** 릴리스마다 변경될 수 있으며 이러한 경우는 모든 구성 가능성이 제어되는 경우에만 정상적으로 처리될 수 있습니다. 이 문서에 설명된 것과 다른 구성을 사용하는 경우 **OpenShift Elasticsearch Operator**와 **Red Hat OpenShift Logging Operator**가 차이를 조정하므로 변경한 내용이 사라집니다. **Operator**는 원래 기본적으로 모든 항목을 정의된 상태로 되돌립니다.

참고

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

4.2.2. 로깅 수집기 Pod 보기

Fluentd 로깅 수집기 **Pod**와 실행 중인 해당 노드를 볼 수 있습니다. **Fluentd** 로깅 수집기 **Pod**는 **openshift-logging** 프로젝트에서만 실행됩니다.

프로세스

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

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

출력 예

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

4.2.3. 로그 수집기 CPU 및 메모리 제한 구성

로그 수집기는 CPU 및 메모리 제한을 모두 조정할 수 있습니다.

프로세스

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

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

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

...

spec:
  collection:
    logs:
      fluentd:
        resources:
          limits: 1
            memory: 736Mi
          requests:
            cpu: 100m
            memory: 736Mi
```

1

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

4.2.4. 로그 전달자를 위한 고급 구성

OpenShift Logging에는 **Fluentd** 로그 전달자의 성능을 조정하는 데 사용할 수 있는 여러 **Fluentd** 매개변수가 포함됩니다. 이러한 매개변수를 사용하여 다음 **Fluentd** 동작을 변경할 수 있습니다.

- 청크 및 청크 버퍼 크기
- 청크 플러시 동작
- 청크 전달 재시도 동작

Fluentd는 *청크*라는 단일 **blob**에서 로그 데이터를 수집합니다. **Fluentd**가 청크를 생성할 때 청크는 *스테이지*에 있는 것으로 간주되어 청크가 데이터로 채워집니다. 청크가 가득 차면 **Fluentd**는 청크를 *큐*로 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. 기본적으로 **Fluentd**는 청크 플러싱을 무기한 재 시도합니다. **OpenShift Container Platform**에서는 무제한 재시도 동작을 변경할 수 없습니다.

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

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

ClusterLogging 사용자 정의 리소스(CR)에서 다음 매개변수를 사용하여 청크 및 플러시 동작을 구성할 수 있습니다. 그러면 Fluentd에서 사용할 수 있도록 매개변수가 Fluentd 구성 맵에 자동으로 추가됩니다.

참고

이러한 매개변수는 다음과 같습니다.

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

표 4.1. 고급 Fluentd 구성 매개변수

매개변수	설명	기본
chunkLimitSize	각 청크의 최대 크기입니다. Fluentd는 이 크기에 도달하면 청크에 데이터 쓰기를 중지합니다. 그런 다음 Fluentd는 청크를 큐로 보내고 새 청크를 엽니다.	**8m**
totalLimitSize	스테이지와 큐의 총 크기인 버퍼의 최대 크기입니다. 버퍼 크기가 이 값을 초과하면 Fluentd는 청크로의 데이터 추가를 중지하고 오류와 함께 실패합니다. 청크에 없는 모든 데이터는 손실됩니다.	**8G**
flushInterval	청크 플러시 간격입니다. **s**(초), **m**(분), **h**(시간) 또는 **d**(일)를 사용할 수 있습니다.	**1s**

매개변수	설명	기본
flushMode	플러시를 수행하는 방법: ● **lazy**: **timekey** 매개 변수를 기반으로 청크를 플러시합니다. **timekey** 매개 변수는 수정할 수 없습니다. ● **interval**: flush**Interval** 매개변수를 기반으로 청크를 플러시합니다. ● **immediate**: 데이터가 청크에 추가된 직후 청크를 플러시합니다.	**간격**
flushThreadCount	청크 플러시를 수행하는 스레드 수입니다. 스레드 수를 늘리면 플러시 처리량이 향상되어 네트워크 대기 시간이 숨겨집니다.	**2**
overflowAction	큐가 가득 찼을 때 청크 동작: ● **throw_exception**: 로그에 표시할 예외를 높입니다. ● **block**: 전체 버퍼 문제가 해결될 때까지 데이터 청크를 중지합니다. ● **drop_oldest_chunk**: 가장 오래된 청크를 삭제하여 새로 들어오는 청크를 수락합니다. 오래된 청크는 새로운 청크보다 가치가 적습니다.	**블록**
retryMaxInterval	**exponential_backoff** 재시도 방법의 최대 시간(초)입니다.	**300s**

매개변수	설명	기본
retryType	플러시 실패 시 재시도 방법: • **exponential_backoff**: 플러시 재시도 간격을 늘립니다. Fluentd는 **retry_max_interval** 매개변수에 도달할 때까지 다음 재시도까지 대기하는 시간을 두 배로 늘립니다. • **periodic**: **retryWait** 매개 변수를 기반으로 플러시를 주기적으로 재시도 합니다.	**exponential_backoff**
retryWait	다음 청크 플러시 전의 시간(초)입니다.	**1s**

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

프로세스

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

```
$ oc edit ClusterLogging instance
```

2. **다음 매개변수를 추가하거나 수정합니다.**

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
  name: instance
  namespace: openshift-logging
spec:
  forwarder:
    fluentd:
      buffer:
        chunkLimitSize: 8m (1)
        flushInterval: 5s (2)
        flushMode: interval (3)
        flushThreadCount: 3 (4)
        overflowAction: throw_exception (5)
        retryMaxInterval: "300s" (6)
        retryType: periodic (7)
```

```
      retryWait: 1s (8)
      totalLimitSize: 32m (9)
...
```

(1) 플러시를 위해 큐에 추가되기 전에 각 청크의 최대 크기를 지정합니다.

(2) 청크 플러시 간격을 지정합니다.

(3) **lazy, interval** 또는 **immediate** 등 청크 플러시를 수행할 방법을 지정합니다.

(4) 청크 플러시에 사용할 스레드 수를 지정합니다.

(5) **throw_exception, block** 또는 **drop_oldest_chunk** 등 큐가 가득 찼을 때의 청크 동작을 지정합니다.

(6) **exponential_backoff** 청크 플러시 방법의 최대 간격(초)을 지정합니다.

(7) 청크 플러시 실패 시 재시도 유형을 **exponential_backoff** 또는 **periodic**으로 지정합니다.

(8) 다음 청크 플러시 전 시간(초)을 지정합니다.

(9) 청크 버퍼의 최대 크기를 지정합니다.

3. **Fluentd Pod**가 재배포되었는지 확인합니다.

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

4. 새 값이 **fluentd** 구성 맵에 있는지 확인합니다.

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

예: fluentd.conf

```
<buffer>
 @type file
 path '/var/lib/fluentd/default'
 flush_mode interval
 flush_interval 5s
 flush_thread_count 3
 retry_type periodic
 retry_wait 1s
 retry_max_interval 300s
 retry_timeout 60m
 queued_chunks_limit_size "#{ENV['BUFFER_QUEUE_LIMIT'] || '32'}"
 total_limit_size 32m
 chunk_limit_size 8m
 overflow_action throw_exception
</buffer>
```

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

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

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

사전 요구 사항

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

```
outputRefs:
- default
```

주의

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

프로세스

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

```
$ oc edit ClusterLogging instance
```

2. **ClusterLogging CR**에서 **logStore, visualization** 스탠자를 제거하십시오.

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

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: "openshift-logging"
spec:
  managementState: "Managed"
  collection:
    logs:
      type: "fluentd"
      fluentd: {}
```

4. **Fluentd Pod**가 재배포되었는지 확인합니다.

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

추가 리소스

- 타사 시스템에 로그 전달

4.3. 로그 저장소 구성

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

다음을 포함하여 로그 저장소를 수정할 수 있습니다.

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

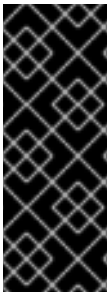
Elasticsearch는 메모리를 많이 사용하는 애플리케이션입니다. ClusterLogging 사용자 정의 리소스에서 달리 지정하지 않는 한 각 Elasticsearch 노드에는 메모리 요청 및 제한 모두에 최소 16G의 메모리가 필요합니다. 초기 OpenShift Container Platform 노드 세트는 Elasticsearch 클러스터를 지원하기에 충분히 크지 않을 수 있습니다. 권장 메모리 이상에서 각 Elasticsearch 노드에 대해 최대 64G까지 실행하려면 OpenShift Container Platform 클러스터에 노드를 추가해야 합니다.

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

4.3.1. 감사 로그를 로그 저장소로 전달

기본적으로 OpenShift Logging은 감사 로그를 내부 OpenShift Container Platform Elasticsearch 로그 저장소에 저장하지 않습니다. 예를 들어 Kibana에서 감사 로그를 볼 수 있도록 이 로그 저장소로 감사 로그를 보낼 수 있습니다.

예를 들어 Kibana에서 감사 로그를 보기 위해 감사 로그를 기본 내부 Elasticsearch 로그 저장소로 보내려면 로그 전달 API를 사용해야 합니다.



중요

내부 **OpenShift Container Platform Elasticsearch** 로그 저장소는 감사 로그를 위한 보안 스토리지를 제공하지 않습니다. 감사 로그를 전달하는 시스템이 조직 및 정부 규정을 준수하고 올바르게 보호되는지 확인합니다. **OpenShift Logging**은 이러한 규정을 준수하지 않습니다.

프로세스

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

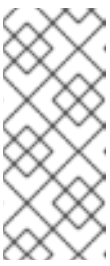
1. **ClusterLogForwarder CR** 오브젝트를 정의하는 **YAML** 파일을 생성하거나 편집합니다.

- 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 CR이 있는 경우 감사 로그의 기본 출력에 파이프라인을 추가합니다. 기본 출력을 정의할 필요가 없습니다. 예를 들면 다음과 같습니다.

```
apiVersion: "logging.openshift.io/v1"
kind: ClusterLogForwarder
metadata:
  name: instance
  namespace: openshift-logging
spec:
  outputs:
   - name: elasticsearch-insecure
     type: "elasticsearch"
     url: http://elasticsearch-insecure.messaging.svc.cluster.local
     insecure: true
   - name: elasticsearch-secure
     type: "elasticsearch"
     url: https://elasticsearch-secure.messaging.svc.cluster.local
     secret:
       name: es-audit
   - name: secureforward-offcluster
     type: "fluentdForward"
     url: https://secureforward.offcluster.com:24224
     secret:
       name: secureforward
  pipelines:
   - name: container-logs
     inputRefs:
     - application
     outputRefs:
     - secureforward-offcluster
   - name: infra-logs
     inputRefs:
     - infrastructure
     outputRefs:
     - elasticsearch-insecure
   - name: audit-logs
     inputRefs:
     - audit
     outputRefs:
     - elasticsearch-secure
     - default (1)
```

(1)

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

추가 리소스

- Log Forwarding API에 대한 자세한 내용은 Log Forwading API를 사용하여 로그 전달을 참조하십시오.

4.3.2. 로그 보존 시간 구성

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

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

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

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

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

사전 요구 사항

- OpenShift Logging 및 OpenShift Elasticsearch Operator가 설치되어 있어야 합니다.

절차

로그 보존 시간을 구성하려면 다음을 수행합니다.

1. retentionPolicy 매개변수를 추가하거나 수정하려면 ClusterLogging CR을 편집합니다.

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

```
spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    retentionPolicy: 1
      application:
        maxAge: 1d
      infra:
        maxAge: 7d
      audit:
        maxAge: 7d
    elasticsearch:
      nodeCount: 3
...
```

1

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

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

예를 들어 **Red Hat OpenShift Logging Operator**가 **8**시간마다 인프라 로그의 활성 인덱스를 롤오버하는 설정이 포함된 보존 정책을 구성하기 위해 다음 **Elasticsearch CR**을 업데이트했고, 롤오버된 인덱스는 롤오버 후 **7**일이 지나면 삭제됩니다. **OpenShift Container Platform**은 **15**분마다 인덱스를 롤오버해야 하는지 확인합니다.

```
apiVersion: "logging.openshift.io/v1"
kind: "Elasticsearch"
metadata:
  name: "elasticsearch"
spec:
...
  indexManagement:
    policies: 1
      - name: infra-policy
        phases:
          delete:
            minAge: 7d 2
          hot:
            actions:
              rollover:
                maxAge: 8h 3
        pollInterval: 15m 4
...
```

1

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

2

OpenShift Container Platform이 롤오버된 인덱스를 삭제하는 경우 이 설정은 **ClusterLogging CR**에서 설정한 **maxAge**입니다.

3

인덱스를 롤오버할 때 고려해야 할 **OpenShift Container Platform**의 인덱스 수명입니다. 이 값은 **ClusterLogging CR**에서 설정한 **maxAge**에서 결정됩니다.

4

OpenShift Container Platform에서 인덱스를 롤오버해야 하는지 확인하는 경우 이 설정은 기본값이며 변경할 수 없습니다.

참고

Elasticsearch CR 수정은 지원되지 않습니다. 보존 정책에 대한 모든 변경은 **ClusterLogging CR**에서 수행해야 합니다.

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

```
$ oc get cronjob
```

출력 예

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

4.3.3. 로그 저장소에 대한 CPU 및 메모리 요청 구성

각 구성 요소 사양을 통해 **CPU** 및 메모리 요청을 조정할 수 있습니다. **OpenShift Elasticsearch 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 **Pod**에 기본 **16Gi** 이상이 할당되어 있어야 합니다. 가급적 **Pod**당 최대 **64Gi**를 할당해야 합니다.

사전 요구 사항

- **OpenShift Logging** 및 **Elasticsearch**가 설치되어 있어야 합니다.

프로세스

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

```
$ oc edit ClusterLogging instance
```

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
....
spec:
    logStore:
      type: "elasticsearch"
      elasticsearch: 1
        resources:
          limits: 2
            memory: "32Gi"
          requests: 3
            cpu: "1"
            memory: "16Gi"
        proxy: 4
          resources:
            limits:
              memory: 100Mi
            requests:
              memory: 100Mi
```

1

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

2

포드에서 사용할 수 있는 최대 리소스 양입니다.

3

Pod를 예약하는 데 필요한 최소 리소스입니다.

4

필요에 따라 Elasticsearch 프록시에 대한 CPU 및 메모리 요청을 지정합니다. 이러한 값을 비워 두면 OpenShift Elasticsearch Operator는 대부분의 배포에 충분한 기본값을 설정합니다. 기본값은 메모리 요청 시 256Mi이고 CPU 요청 시 100m입니다.

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

예를 들면 다음과 같습니다.

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

1

리소스의 최대 양입니다.

2

필요한 최소량.

쿠버네티스는 일반적으로 노드 구성을 준수하며 Elasticsearch가 지정된 제한을 사용하도록 허용하지 않습니다. requests 및 limits에 대해 동일한 값을 설정하면 노드에 사용 가능한 메모리가 있다고 가정하고 Elasticsearch가 원하는 메모리를 사용할 수 있습니다.

4.3.4. 로그 저장소에 대한 복제 정책 구성

Elasticsearch shard가 클러스터의 데이터 노드에 복제되는 방법을 정의할 수 있습니다.

사전 요구 사항

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

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

```
$ oc edit clusterlogging instance
```

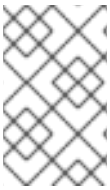
```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
 name: "instance"

....

spec:
 logStore:
  type: "elasticsearch"
  elasticsearch:
   redundancyPolicy: "SingleRedundancy" 1
```

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

- FullRedundancy. Elasticsearch는 각 인덱스의 기본 shard를 모든 데이터 노드에 완전히 복제합니다. 이 방법은 안전성이 가장 높지만 필요한 디스크 양이 가장 많고 성능이 가장 낮습니다.
- MultipleRedundancy. Elasticsearch는 각 인덱스의 기본 shard를 데이터 노드의 절반으로 완전히 복제합니다. 이 방법은 안전성과 성능 사이의 균형이 우수합니다.
- SingleRedundancy. Elasticsearch는 각 인덱스에 대해 기본 shard의 사본 하나를 만듭니다. 두 개 이상의 데이터 노드가 존재하는 한 항상 로그를 사용할 수 있고 복

구할 수 있습니다. 5개 이상의 노드를 사용하는 경우 **MultipleRedundancy**보다 성능이 향상됩니다. 단일 **Elasticsearch** 노드 배포에는 이 정책을 적용할 수 없습니다.

- **ZeroRedundancy**. **Elasticsearch**는 기본 **shard**의 사본을 만들지 않습니다. 노드가 다운되거나 실패하는 경우 로그를 사용할 수 없거나 로그가 손실될 수 있습니다. 안전보다 성능이 더 중요하거나 자체 디스크/**PVC** 백업/복원 전략을 구현한 경우 이 모드를 사용합니다.



참고

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

4.3.5. Elasticsearch Pod 축소

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

축소하는 경우 **Pod**를 한 번에 하나씩 축소하고 클러스터에서 **shard**와 복제본의 균형을 다시 조정할 수 있어야 합니다. **Elasticsearch** 상태가 **green**으로 돌아가면 다른 **Pod**에서 축소할 수 있습니다.

참고

Elasticsearch 클러스터가 **ZeroRedundancy**로 설정된 경우 **Elasticsearch Pod**를 축소해서는 안 됩니다.

4.3.6. 로그 저장소에 대한 영구 스토리지 구성

Elasticsearch에는 영구 스토리지가 필요합니다. 스토리지가 빠를수록 **Elasticsearch** 성능이 빨라집니다.

주의

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

사전 요구 사항

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

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

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
# ...
spec:
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      storage:
        storageClassName: "gp2"
        size: "200G"
```

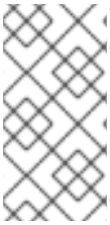
이 예에서는 클러스터의 각 데이터 노드가 AWS General Purpose SSD(gp2) 스토리지 "200G"를 요청하는 영구 볼륨 클레임에 바인딩되도록 지정합니다.

참고

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

4.3.7. emptyDir 스토리지에 대한 로그 저장소 구성

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



참고

emptyDir을 사용할 때 로그 스토리지가 다시 시작되거나 재배포되면 데이터가 손실됩니다.

사전 요구 사항

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

1. emptyDir을 지정하려면 ClusterLogging CR을 편집합니다.

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

4.3.8. Elasticsearch 롤링 클러스터 재시작 수행

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

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

사전 요구 사항

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

클러스터를 롤링 재시작하려면 다음을 수행합니다.

1. **openshift-loggin 프로젝트로 변경합니다.**

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

2. **Elasticsearch pod의 이름을 가져옵니다.**

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

3. **Fluentd Pod를 축소하여 Elasticsearch로 새 로그 전송을 중지합니다.**

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

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

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

예를 들면 다음과 같습니다.

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

출력 예

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

5. **OpenShift Container Platform es_util 도구를 사용하여 의도적으로 노드를 중단할 때 shard 밸런싱을 방지합니다.**

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

예를 들면 다음과 같습니다.

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

출력 예

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

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

a. 기본적으로 **OpenShift Container Platform Elasticsearch** 클러스터는 노드에 대한 롤아웃을 차단합니다. 다음 명령을 사용하여 롤아웃을 허용하고 **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
```

새 **Pod**가 배포되었습니다. **Pod**에 컨테이너가 준비되면 다음 배포로 이동할 수 있습니다.

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

-

출력 예

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

b. 배포가 완료되면 롤아웃을 허용하지 않도록 **Pod**를 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 ConfigMap**을 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. Fluentd Pod를 확장하여 Elasticsearch에 새 로그를 전송합니다.

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

4.3.9. 로그 저장소 서비스를 경로로 노출

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

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

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

내부에서는 다음 명령 중 하나로 얻을 수 있는 로그 저장소 클러스터 IP를 사용하여 로그 저장소 서비스에 액세스할 수 있습니다.

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

출력 예

```
172.30.183.229
```

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

출력 예

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

다음과 유사한 명령을 사용하여 클러스터 IP 주소를 확인할 수 있습니다.

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

출력 예

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

사전 요구 사항

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

- 로그에 액세스하려면 프로젝트에 액세스할 수 있어야 합니다.

프로세스

로그 저장소를 외부에 노출하려면 다음을 수행합니다.

1. openshift-loggin 프로젝트로 변경합니다.

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

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

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

출력 예

```
admin-ca
```

3. 로그 저장소 서비스의 경로를 YAML 파일로 생성합니다.

   a. 다음을 사용하여 YAML 파일을 생성합니다.

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

1

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

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

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

c. 경로를 생성합니다.

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

출력 예

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

4. Elasticsearch 서비스가 노출되어 있는지 확인합니다.

a. 요청에 사용할 이 서비스 계정의 토큰을 가져옵니다.

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

b. 생성한 elasticsearch 경로를 환경 변수로 설정합니다.

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

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

```
curl -tlsv1.2 --insecure -H "Authorization: Bearer ${token}" "https://${routeES}"
```

응답은 다음과 유사하게 나타납니다.

출력 예

```
{
  "name" : "elasticsearch-cdm-i40ktba0-1",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "0eY-tJzcR3KOdpgeMJo-MQ",
  "version" : {
  "number" : "6.8.1",
  "build_flavor" : "oss",
  "build_type" : "zip",
  "build_hash" : "Unknown",
  "build_date" : "Unknown",
  "build_snapshot" : true,
  "lucene_version" : "7.7.0",
  "minimum_wire_compatibility_version" : "5.6.0",
  "minimum_index_compatibility_version" : "5.0.0"
},
  "<tagline>" : "<for search>"
}
```

4.4. 로그 시각화 프로그램 구성

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

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

4.4.1. CPU 및 메모리 제한 구성

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

프로세스

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

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

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

...

spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      resources: 1
        limits:
          memory: 16Gi
        requests:
          cpu: 200m
          memory: 16Gi
      storage:
        storageClassName: "gp2"
        size: "200G"
      redundancyPolicy: "SingleRedundancy"
  visualization:
    type: "kibana"
    kibana:
      resources: 2
        limits:
          memory: 1Gi
        requests:
          cpu: 500m
          memory: 1Gi
      proxy:
        resources: 3
          limits:
            memory: 100Mi
          requests:
            cpu: 100m
            memory: 100Mi
      replicas: 2
  collection:
    logs:
      type: "fluentd"
      fluentd:
        resources: 4
          limits:
            memory: 736Mi
          requests:
            cpu: 200m
            memory: 736Mi
```

1

2 3

필요에 따라 로그 시각화 프로그램에 대한 CPU 및 메모리 제한 및 요청을 지정합니다.

4

필요에 따라 로그 수집기에 대한 CPU 및 메모리 제한 및 요청을 지정합니다.

4.4.2. 로그 시각화 프로그램 노드의 확장성 중복

중복성에 대해 로그 시각화 프로그램을 호스팅하는 Pod를 확장할 수 있습니다.

프로세스

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

```
$ oc edit ClusterLogging instance
```

```
$ oc edit ClusterLogging instance

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

....

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

1

Kibana 노드의 수를 지정합니다.

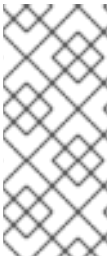
4.5. OPENSHIFT LOGGING 스토리지 구성

Elasticsearch는 메모리를 많이 사용하는 애플리케이션입니다. 기본 OpenShift Logging 설치는 메모리 요청 및 메모리 제한 모두에 16G 메모리를 배포합니다. 초기 OpenShift Container Platform 노드 세트는 Elasticsearch 클러스터를 지원하기에 충분히 크지 않을 수 있습니다. 권장 메모리 이상으로 실행하

려면 OpenShift Container Platform 클러스터에 노드를 추가해야 합니다. 각 Elasticsearch 노드는 더 낮은 메모리 설정으로 작동할 수 있지만 프로덕션 환경에는 권장되지 않습니다.

4.5.1. OpenShift Logging 및 OpenShift Container Platform에 대한 스토리지 고려 사항

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



참고

영구 스토리지에 로컬 볼륨을 사용하는 경우 LocalVolume 개체에서 volumeMode: 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 journal 및 /var/log/containers/의 모든 로그를 Elasticsearch에 제공합니다.

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

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

참고

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

4.5.2. 추가 리소스

- 로그 저장소에 대한 영구 스토리지 구성

4.6. OPENSHIFT LOGGING 구성 요소에 대한 CPU 및 메모리 제한 구성

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

4.6.1. CPU 및 메모리 제한 구성

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

프로세스

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

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

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

...

spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      resources: 1
        limits:
          memory: 16Gi
        requests:
          cpu: 200m
          memory: 16Gi
      storage:
        storageClassName: "gp2"
        size: "200G"
      redundancyPolicy: "SingleRedundancy"
  visualization:
    type: "kibana"
    kibana:
      resources: 2
        limits:
          memory: 1Gi
        requests:
          cpu: 500m
          memory: 1Gi
      proxy:
```

```
      resources: 3
        limits:
          memory: 100Mi
        requests:
          cpu: 100m
          memory: 100Mi
    replicas: 2
collection:
  logs:
    type: "fluentd"
    fluentd:
      resources: 4
        limits:
          memory: 736Mi
        requests:
          cpu: 200m
          memory: 736Mi
```

1

필요에 따라 로그 저장소에 대한 CPU 및 메모리 제한 및 요청을 지정합니다. Elasticsearch의 경우 요청 값과 제한 값을 모두 조정해야 합니다.

2 3

필요에 따라 로그 시각화 프로그램에 대한 CPU 및 메모리 제한 및 요청을 지정합니다.

4

필요에 따라 로그 수집기에 대한 CPU 및 메모리 제한 및 요청을 지정합니다.

4.7. 허용 오차를 사용하여 OPENSHIFT LOGGING POD 배치 제어

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

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

key는 최대 253자의 문자열이고 value은 최대 63자의 문자열입니다. 문자열은 문자 또는 숫자로 시작해야 하며 문자, 숫자, 하이픈, 점 및 밑줄을 포함할 수 있습니다.

허용 오차가 있는 샘플 OpenShift Logging CR

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

...

spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      tolerations: 1
      - key: "logging"
        operator: "Exists"
        effect: "NoExecute"
        tolerationSeconds: 6000
      resources:
        limits:
          memory: 16Gi
        requests:
          cpu: 200m
          memory: 16Gi
      storage: {}
      redundancyPolicy: "ZeroRedundancy"
  visualization:
    type: "kibana"
    kibana:
      tolerations: 2
      - key: "logging"
        operator: "Exists"
        effect: "NoExecute"
        tolerationSeconds: 6000
      resources:
        limits:
          memory: 2Gi
        requests:
          cpu: 100m
          memory: 1Gi
      replicas: 1
  collection:
    logs:
      type: "fluentd"
      fluentd:
        tolerations: 3
        - key: "logging"
          operator: "Exists"
          effect: "NoExecute"
          tolerationSeconds: 6000
        resources:
          limits:
            memory: 2Gi
```

```
      requests:
        cpu: 100m
        memory: 1Gi
```

1

이 허용 오차는 **Elasticsearch Pod**에 추가됩니다.

2

이 허용 오차는 **Kibana Pod**에 추가됩니다.

3

이 허용 오차는 로깅 수집기 **Pod**에 추가됩니다.

4.7.1. 허용 오차를 사용하여 로그 저장소 Pod 배치 제어

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

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

기본적으로 로그 저장소 **Pod**에는 다음과 같은 허용 오차가 있습니다.

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

사전 요구 사항

- **OpenShift Logging** 및 **Elasticsearch**가 설치되어 있어야 합니다.

프로세스

1. 다음 명령을 사용하여 **OpenShift Logging Pod**를 예약하려는 노드에 **taint**를 추가합니다.

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

예를 들면 다음과 같습니다.

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

이 예에서는 키 **elasticsearch**, 값 **node** 및 **taint** 효과 **NoExecute**로 **node1**에 **taint**를 배치합니다. **NoExecute** 효과가 있는 노드는 **taint**와 일치하는 **Pod**만 스케줄링하고 일치하지 않는 기존 **Pod**는 제거합니다.

2. **Elasticsearch Pod**에 대한 허용 오차를 구성하려면 **ClusterLogging CR**의 **logstore** 섹션을 편집합니다.

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

1. 노드에 추가한 키를 지정합니다.

2. 노드에 **elasticsearch** 키의 **taint**가 존재할 것을 요구하도록 **Exists Operator**를 지정합니다.

3. **NoExecute** 효과를 지정합니다.

4. 선택적으로 **tolerationSeconds** 매개변수를 지정하여 **Pod**가 제거되기 전까지 노드에 바인딩되는 시간을 설정합니다.

이 허용 오차는 **oc adm taint** 명령으로 생성된 **taint**와 일치합니다. 이 허용 오차가 있는 **Pod**를 **node1**에 예약할 수 있습니다.

4.7.2. 허용 오차를 사용하여 로그 시각화 프로그램 Pod 배치 제어

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

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

사전 요구 사항

- **OpenShift Logging** 및 **Elasticsearch**가 설치되어 있어야 합니다.

프로세스

1. 다음 명령을 사용하여 로그 시각화 프로그램 **Pod**를 예약하려는 노드에 **taint**를 추가합니다.

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

예를 들면 다음과 같습니다.

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

이 예에서는 키 **kibana**, 값 **node** 및 **taint** 효과 **NoExecute**로 **node1**에 **taint**를 배치합니다. **NoExecute taint** 효과를 사용해야 합니다. **NoExecute**는 **taint**와 일치하는 **Pod**만 스케줄링하고 일치하지 않는 기존 **Pod**는 제거합니다.

2. **Kibana Pod**에 대한 허용 오차를 구성하려면 **ClusterLogging CR**의 **visualization** 섹션을 편집합니다.

```
visualization:
  type: "kibana"
  kibana:
    tolerations:
```

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

1 노드에 추가한 키를 지정합니다.

2 key/value/effect 매개변수가 일치할 것을 요구하도록 Exists Operator를 지정합니다.

3 NoExecute 효과를 지정합니다.

4 선택적으로 tolerationSeconds 매개변수를 지정하여 Pod가 제거되기 전까지 노드에 바인딩되는 시간을 설정합니다.

이 허용 오차는 oc adm taint 명령으로 생성된 taint와 일치합니다. 이 허용 오차가 있는 Pod는 node1에 스케줄링할 수 있습니다.

4.7.3. 허용 오차를 사용하여 로그 수집기 Pod 배치 제어

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

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

기본적으로 로깅 수집기 Pod에는 다음과 같은 허용 오차가 있습니다.

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

사전 요구 사항

- **OpenShift Logging** 및 **Elasticsearch**가 설치되어 있어야 합니다.

프로세스

1. 다음 명령을 사용하여 로깅 수집기 **Pod**에서 로깅 수집기 **Pod**를 스케줄링할 노드에 **taint**를 추가합니다.

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

예를 들면 다음과 같습니다.

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

이 예에서는 키 **collector**, 값 **node** 및 **taint** 효과 **NoExecute**로 **node1**에 **taint**를 배치합니다. **NoExecute taint** 효과를 사용해야 합니다. **NoExecute**는 **taint**와 일치하는 **Pod**만 스케줄링하고 일치하지 않는 기존 **Pod**는 제거합니다.

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

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

(1) 노드에 추가한 키를 지정합니다.

(2) **key/value/effect** 매개변수가 일치할 것을 요구하도록 **Exists Operator**를 지정합니다.

(3)

4

선택적으로 **tolerationSeconds** 매개변수를 지정하여 **Pod**가 제거되기 전까지 노드에 바인딩되는 시간을 설정합니다.

이 허용 오차는 **oc adm taint** 명령으로 생성된 **taint**와 일치합니다. 이 허용 오차가 있는 **Pod**는 **node1**에 스케줄링할 수 있습니다.

4.7.4. 추가 리소스

- 노드 테인트를 사용하여 **Pod** 배치 제어

4.8. 노드 선택기로 OPENSHIFT LOGGING 리소스 이동

노드 선택기를 사용하여 **Elasticsearch, Kibana Pod**를 다른 노드에 배포할 수 있습니다.

4.8.1. OpenShift Logging 리소스 이동

Elasticsearch 및 **Kibana**와 같은 **OpenShift Logging** 구성 요소 용 **Pod**를 다른 노드에 배포하도록 **OpenShift Logging Operator**를 구성할 수 있습니다. 설치된 위치에서 **Cluster Logging Operator Pod**를 이동할 수 없습니다.

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

사전 요구 사항

- **OpenShift Logging** 및 **Elasticsearch**가 설치되어 있어야 합니다. 이러한 기능은 기본적으로 설치되지 않습니다.

프로세스

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

```
$ oc edit ClusterLogging instance
```

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

...

spec:
  collection:
    logs:
      fluentd:
        resources: null
      type: fluentd
  logStore:
    elasticsearch:
      nodeCount: 3
      nodeSelector: 1
        node-role.kubernetes.io/infra: ''
      tolerations:
      - effect: NoSchedule
        key: node-role.kubernetes.io/infra
        value: reserved
      - effect: NoExecute
        key: node-role.kubernetes.io/infra
        value: reserved
      redundancyPolicy: SingleRedundancy
      resources:
        limits:
          cpu: 500m
          memory: 16Gi
        requests:
          cpu: 500m
          memory: 16Gi
      storage: {}
    type: elasticsearch
  managementState: Managed
  visualization:
    kibana:
      nodeSelector: 2
        node-role.kubernetes.io/infra: ''
      tolerations:
      - effect: NoSchedule
        key: node-role.kubernetes.io/infra
        value: reserved
      - effect: NoExecute
        key: node-role.kubernetes.io/infra
        value: reserved
      proxy:
        resources: null
      replicas: 1
      resources: null
    type: kibana

...
```

1 2

검증

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

예를 들면 다음과 같습니다.

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

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

출력 예

```
NAME                     READY   STATUS    RESTARTS   AGE   IP            NODE
NOMINATED NODE   READINESS GATES
kibana-5b8bdf44f9-ccpq9   2/2     Running   0          27s   10.129.2.18   ip-10-0-147-79.us-
east-2.compute.internal   <none>           <none>
```

- **Kibana Pod**를 전용 인프라 노드인 **ip-10-0-139-48.us-east-2.compute.internal** 노드로 이동하려는 경우 다음을 실행합니다.

```
$ oc get nodes
```

출력 예

```
NAME                                         STATUS   ROLES        AGE   VERSION
ip-10-0-133-216.us-east-2.compute.internal   Ready    master       60m   v1.21.0
ip-10-0-139-146.us-east-2.compute.internal   Ready    master       60m   v1.21.0
ip-10-0-139-192.us-east-2.compute.internal   Ready    worker       51m   v1.21.0
ip-10-0-139-241.us-east-2.compute.internal   Ready    worker       51m   v1.21.0
ip-10-0-147-79.us-east-2.compute.internal    Ready    worker       51m   v1.21.0
ip-10-0-152-241.us-east-2.compute.internal   Ready    master       60m   v1.21.0
ip-10-0-139-48.us-east-2.compute.internal    Ready    infra        51m   v1.21.0
```

노드에는 **node-role.kubernetes.io/infra : ''** 레이블이 있음에 유의합니다.

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

출력 예

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

- Kibana pod를 이동하려면 **ClusterLogging CR**을 편집하여 노드 선택기를 추가합니다.

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

...

spec:

...

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

1

- CR을 저장하면 현재 Kibana pod가 종료되고 새 pod가 배포됩니다.

```
$ oc get pods
```

출력 예

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

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

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

출력 예

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

- 잠시 후 원래 Kibana pod가 제거됩니다.

```
$ oc get pods
```

출력 예

```
NAME                                        READY   STATUS    RESTARTS   AGE
cluster-logging-operator-84d98649c4-zb9g7   1/1     Running   0          30m
elasticsearch-cdm-hwv01pf7-1-56588f554f-kpmlg   2/2     Running   0          29m
elasticsearch-cdm-hwv01pf7-2-84c877d75d-75wqj   2/2     Running   0          29m
elasticsearch-cdm-hwv01pf7-3-f5d95b87b-4nx78    2/2     Running   0          29m
fluentd-42dzz                               1/1     Running   0          29m
fluentd-d74rq                               1/1     Running   0          29m
fluentd-m5vr9                               1/1     Running   0          29m
fluentd-nkxl7                               1/1     Running   0          29m
fluentd-pdvqb                               1/1     Running   0          29m
fluentd-tflh6                               1/1     Running   0          29m
kibana-7d85dcffc8-bfpfp                     2/2     Running   0          62s
```

4.9. SYSTEMD-JOURNALD 및 FLUENTD 구성

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

저널이 항목을 손실하지 않도록 RateLimitIntervalSec=30s 및 RateLimitBurst = 10000(또는 필요한 경우 더 높음)을 설정하는 것이 좋습니다.

4.9.1. OpenShift Logging을 위한 systemd-journald 구성

프로젝트를 확장할 때 기본 로깅 환경을 조정해야 할 수도 있습니다.

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

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

절차

1. 필요한 설정과 함께 /etc/systemd/journald.conf 파일을 포함하는 Butane 구성 파일 40-worker-custom-journald.bu를 만듭니다.

참고

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

```
variant: openshift
version: 4.8.0
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) journal.conf 파일에 대한 권한을 설정합니다. 0644 권한을 설정하는 것이 좋습니다.

(2) 로그를 파일 시스템에 쓰기 전에 압축할지 여부를 지정합니다. 메시지를 압축하려면 yes를 지정하고 압축하지 않으려면 no를 지정합니다. 기본값은 yes입니다.

(3) 로그 메시지를 전달할지 여부를 구성합니다. 각각에 대해 기본값은 no입니다. 다음을 지정합니다.

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

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.

- **/var/log/journal/**에서 메모리에 로그를 저장하기 위한 **volatile**입니다.
- **/var/log/journal/**의 디스크에 로그를 저장하기 위한 **persistent**입니다. **systemd**는 디렉토리가 없는 경우 디렉토리를 생성합니다.
- 디렉토리가 존재하는 경우 **/var/log/journal/**에 로그를 저장하기 위한 **auto**입니다. 존재하지 않는 경우 **systemd**는 **/run/systemd/journal**에 로그를 임시 저장합니다.
- 로그를 저장하지 않는 **none**입니다. **systemd**는 모든 로그를 삭제합니다.

7

8

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

9

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

10

/var/log/journal에 지속적으로 저장된 개별 저널 파일의 최대 크기를 지정합니다. 기본값은 **10M**입니다.

참고

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

시스템 설정에 대한 자세한 내용은 https://www.freedesktop.org/software/systemd/man/journald.conf.html을 참조하십시오. 해당 페이지에 나열된 기본 설정은 **OpenShift Container Platform**에 적용되지 않을 수 있습니다.

2. **Butane**을 사용하여 노드로 전달할 구성이 포함된 **MachineConfig** 개체 파일 **40-worker-custom-journald.yaml**을 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 **MachineConfig**를 감지하고 새로운 **rendered-worker-<hash>** 버전을 생성합니다.

4. 각 노드에 새로 렌더링된 구성의 롤아웃 상태를 모니터링합니다.

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

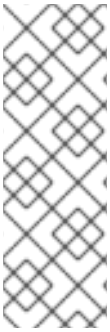
출력 예

```
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
```

4.10. 유지보수 및 지원

4.10.1. 지원되지 않는 구성 정보

지원되는 **OpenShift Logging** 구성 방법은 이 설명서에 설명된 옵션을 사용하여 구성하는 것입니다. 다른 구성은 지원되지 않으므로 사용하지 마십시오. 구성 패러다임은 **OpenShift Container Platform** 릴리스마다 변경될 수 있으며 이러한 경우는 모든 구성 가능성이 제어되는 경우에만 정상적으로 처리될 수 있습니다. 이 문서에 설명된 것과 다른 구성을 사용하는 경우 **OpenShift Elasticsearch Operator**와 **Red Hat OpenShift Logging Operator**가 차이를 조정하므로 변경한 내용이 사라집니다. **Operator**는 원래 기본적으로 모든 항목을 정의된 상태로 되돌립니다.



참고

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

4.10.2. 지원되지 않는 로깅 구성

다음 구성 요소를 수정하려면 **Red Hat OpenShift Logging Operator**를 관리되지 않음 상태로 설정해야 합니다.

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

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

- Elasticsearch 배포 파일.

명시적으로 지원되지 않는 경우는 다음과 같습니다.

- 기본 로그 회전 구성. 기본 로그 회전 구성을 수정할 수 없습니다.
- 수집된 로그 위치 구성. 로그 수집기 출력 파일의 위치는 기본적으로 /var/log/fluentd/fluentd.log입니다.
- 제한 로그 수집. 로그 수집기에서 로그를 읽는 속도를 조절할 수 없습니다.
- 환경 변수를 사용하여 로깅 수집기 구성. 환경 변수를 사용하여 로그 수집기를 수정할 수 없습니다.
- 로그 수집기에서 로그를 정규화하는 방법 구성. 기본 로그 정규화를 수정할 수 없습니다.

4.10.3. 관리되지 않는 Operator에 대한 지원 정책

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

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

다음과 같은 방법으로 Operator를 Unmanaged 상태로 설정할 수 있습니다.

- 개별 Operator 구성

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

  managementState 매개변수를 Unmanaged로 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 true로 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.

5장. 리소스의 로그 보기

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

참고

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

5.1. 리소스 로그 보기

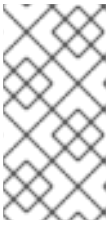
OpenShift CLI(oc) 및 웹 콘솔에서 다양한 리소스의 로그를 볼 수 있습니다. 로그는 로그의 말미 또는 끝에서 읽습니다.

사전 요구 사항

- **OpenShift CLI(oc)**에 액세스합니다.

프로세스(UI)

1. **OpenShift Container Platform** 콘솔에서 워크로드 → **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: 출력이 로그에 기록되는 내용을 따르도록 지정합니다.

<pod_name>

pod 이름을 지정합니다.

<container_name>

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

예를 들면 다음과 같습니다.

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

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

로그 파일의 내용이 출력됩니다.

- 특정 리소스의 로그를 확인합니다.

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

1 리소스 유형 및 이름을 지정합니다.

예를 들면 다음과 같습니다.

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

로그 파일의 내용이 출력됩니다.

6장. KIBANA를 사용하여 클러스터 로그 보기

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

로그 시각화 프로그램을 사용하면 데이터로 다음을 수행할 수 있습니다.

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

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

참고

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

6.1. KIBANA 인덱스 패턴 정의

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

사전 요구 사항

- Kibana에서 인프라 및 감사 인덱스를 보려면 사용자에게 cluster-admin 역할이나 cluster-reader 역할 또는 두 역할이 모두 있어야 합니다. 기본 kubeadmin 사용자에게는 이러한 인덱스를 나열할 수 있는 적절한 권한이 있습니다.

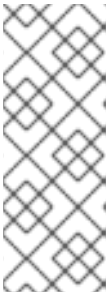
  default, kube-, openshift- 프로젝트에서 Pod와 로그를 볼 수 있다면 이러한 인덱스에 액세스할 수 있어야 합니다. 다음 명령을 사용하여 현재 사용자에게 적절한 권한이 있는지 확인할 수

있습니다.

```
$ oc auth can-i get pods/log -n <project>
```

출력 예

```
yes
```



참고

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

프로세스

Kibana에서 인덱스 패턴을 정의하고 시각화를 생성하려면 다음을 수행합니다.

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

2. 관리 → 인덱스 패턴 → 인덱스 패턴 생성을 클릭하여 **Kibana** 인덱스 패턴을 생성합니다.

- 각 사용자는 프로젝트의 로그를 보려면 **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 로그인할 때 인덱스 패턴을 생성해야 합니다.

3. 새로운 인덱스 패턴에서 **Kibana** 시각화를 생성합니다.

6.2. KIBANA에서 클러스터 로그 보기

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

사전 요구 사항

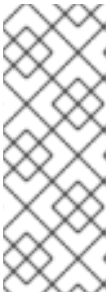
- **OpenShift Logging** 및 **Elasticsearch**가 설치되어 있어야 합니다.
- **Kibana** 인덱스 패턴이 있어야 합니다.
- **Kibana**에서 인프라 및 감사 인덱스를 보려면 사용자에게 **cluster-admin** 역할이나 **cluster-reader** 역할 또는 두 역할이 모두 있어야 합니다. 기본 **kubeadmin** 사용자에게는 이러한 인덱스를 나열할 수 있는 적절한 권한이 있습니다.

  default, kube-, openshift- 프로젝트에서 **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/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.

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. **Kibana**에서 검색을 클릭합니다.

4. 왼쪽 상단 드롭다운 메뉴에서 생성한 인덱스 패턴(**app**, **audit** 또는 **infra**)을 선택합니다.

   로그 데이터가 타임스탬프가 있는 문서로 표시됩니다.

5. 타임스탬프가 있는 문서 중 하나를 확장합니다.

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

예 6.1. Kibana의 샘플 인프라 로그 항목

```
{
  "_index": "infra-000001",
  "_type": "_doc",
  "_id": "YmJmYTBlNDkZTRmLTliMGQtMjE3NmFiOGUyOWM3",
  "_version": 1,
  "_score": null,
```

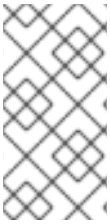
```
  "_source": {
    "docker": {
      "container_id":
"f85fa55bbef7bb783f041066be1e7c267a6b88c4603dfce213e32c1"
    },
    "kubernetes": {
      "container_name": "registry-server",
      "namespace_name": "openshift-marketplace",
      "pod_name": "redhat-marketplace-n64gc",
      "container_image": "registry.redhat.io/redhat/redhat-marketplace-index:v4.7",
      "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": "YmJmYTBlNDktMDMGQtMjE3NmFiOGUyOWM3",
    "openshift": {
      "labels": {
        "logging": "infra"
      }
    }
  },
  "fields": {
    "@timestamp": [
      "2020-09-23T20:47:03.422Z"
    ],
    "pipeline_metadata.collector.received_at": [
      "2020-09-23T20:47:15.007Z"
    ]
  },
  "sort": [
```

```
      1600894023422
    ]
  }
```

7장. 외부 타사 로깅 시스템으로 로그 전달

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

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



참고

감사 로그를 기본 내부 **Elasticsearch** 로그 저장소로 보내려면 로그 저장소에 감사 로그 전달에 설명된 대로 **Cluster Log Forwarder**를 사용합니다.

7.1. 타사 시스템으로 로그 전달 정보

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

- *출력*은 사용자가 정의한 로그 데이터의 대상 또는 로그를 보낼 위치입니다. 출력은 다음 유형 중 하나일 수 있습니다.
  - **elasticsearch.** 외부 **Elasticsearch** 인스턴스입니다. **elasticsearch** 출력은 **TLS** 연결을 사용할 수 있습니다.
  - **fluentdForward. Fluentd**를 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.
  - **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.

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

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

- **default.** 내부 **OpenShift Container Platform Elasticsearch** 인스턴스입니다**.** 기본 출력을 구성할 필요는 없습니다**. default** 출력을 구성하는 경우 **default** 출력이 **Red Hat OpenShift Logging Operator**용으로 예약되므로 오류 메시지가 나타납니다**.**

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

- *파이프라인*은 한 로그 유형에서 하나 이상의 출력 또는 전송할 로그로의 간단한 라우팅을 정의합니다**.** 로그 유형은 다음 중 하나입니다**.**

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

  - **infrastructure. openshift*, kube*** 또는 **default** 프로젝트에서 실행되는 **Pod**의 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**.**

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

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

다음을 확인합니다.

- **ClusterLogForwarder CR** 오브젝트가 있는 경우 **default** 출력이 있는 파이프라인이 없으면 로그가 기본 **Elasticsearch** 인스턴스로 전달되지 않습니다.
- 기본적으로 **OpenShift Container Platform** 클러스터 로깅은 컨테이너 및 인프라 로그를 **ClusterLogging** 사용자 정의 리소스에 정의된 기본 내부 **Elasticsearch** 로그 저장소로 보냅니다. 그러나 보안 스토리지를 제공하지 않기 때문에 감사 로그를 내부 저장소로 보내지 않습니다. 이 기본 구성이 요구 사항을 충족하는 경우 **Log Forwarding API**를 구성하지 마십시오.
- 로그 유형에 대한 파이프라인을 정의하지 않으면 정의되지 않은 유형의 로그가 삭제됩니다. 예를 들어 **application** 및 **audit** 유형에 대한 파이프라인을 지정하고 **infrastructure** 유형에 대한 파이프라인을 지정하지 않으면 **infrastructure** 로그가 삭제됩니다.
- **ClusterLogForwarder** 사용자 정의 리소스(CR)에서 여러 유형의 출력을 사용하여 다른 프로토콜을 지원하는 서버에 로그를 보낼 수 있습니다.
- 내부 **OpenShift Container Platform Elasticsearch** 인스턴스는 감사 로그를 위한 보안 스토리지를 제공하지 않습니다. 감사 로그를 전달하는 시스템이 조직 및 정부 규정을 준수하고 올바르게 보호되도록 하는 것이 좋습니다. **OpenShift Logging**은 이러한 규정을 준수하지 않습니다.
- 키 및 시크릿, 서비스 계정, 포트 열기 또는 전역 프록시 구성과 같이 외부 대상에 필요할 수 있는 추가 구성을 생성하고 유지보수할 책임이 있습니다.

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

샘플 로그 전달 출력 및 파이프라인

```
apiVersion: "logging.openshift.io/v1"
kind: ClusterLogForwarder
metadata:
  name: instance 1
```

```
  namespace: openshift-logging (2)
spec:
  outputs:
   - name: elasticsearch-secure (3)
     type: "elasticsearch"
     url: https://elasticsearch.secure.com:9200
     secret:
       name: elasticsearch
   - name: elasticsearch-insecure (4)
     type: "elasticsearch"
     url: http://elasticsearch.insecure.com:9200
   - name: kafka-app (5)
     type: "kafka"
     url: tls://kafka.secure.com:9093/app-topic
  inputs: (6)
   - name: my-app-logs
     application:
       namespaces:
       - my-project
  pipelines:
   - name: audit-logs (7)
     inputRefs:
      - audit
     outputRefs:
      - elasticsearch-secure
      - default
     parse: json (8)
     labels:
       secure: "true" (9)
       datacenter: "east"
   - name: infrastructure-logs (10)
     inputRefs:
      - infrastructure
     outputRefs:
      - elasticsearch-insecure
     labels:
       datacenter: "west"
   - name: my-app (11)
     inputRefs:
      - my-app-logs
     outputRefs:
      - default
   - inputRefs: (12)
      - application
     outputRefs:
      - kafka-app
     labels:
       datacenter: "south"
```

(1)

2

ClusterLogForwarder CR의 네임스페이스는 openshift-logging이어야 합니다.

3

보안 시크릿과 보안 URL을 사용하여 보안 Elasticsearch 출력을 구성합니다.

- 출력을 설명하는 이름입니다.
- 출력 유형: elasticsearch.
- 접두사를 포함하여 유효한 절대 URL인 Elasticsearch 인스턴스의 보안 URL 및 포트입니다.
- TLS 통신을 위해 끝점에서 요구하는 시크릿입니다. openshift-logging 프로젝트에 이 시크릿이 있어야 합니다.

4

안전하지 않은 Elasticsearch 출력에 대한 구성:

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

5

보안 URL을 통한 클라이언트 인증 TLS 통신을 사용하는 Kafka 출력 구성

- 출력을 설명하는 이름입니다.

- 출력 유형: **kafka.**
- 접두사를 포함하여 **Kafka** 브로커의 **URL** 및 포트를 유효한 절대 **URL**로 지정합니다.

6

my-project 네임스페이스에서 애플리케이션 로그를 필터링하기 위한 입력 구성입니다.

7

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.

8

선택 사항: 구조화된 필드에서 **JSON** 오브젝트로 구조화된 **JSON** 로그 항목을 전달할지 여부를 지정합니다. 로그 항목에 유효한 구조화된 **JSON**이 포함되어야 합니다. 그렇지 않으면 **OpenShift Logging**이 **structured** 필드를 제거하고 대신 기본 인덱스인 **app-00000x**로 로그 항목을 보냅니다.

9

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

10

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

11

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

- 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 인증 키

일반적인 주요 유형은 여기에 제공됩니다. 일부 출력 유형에서는 출력별 구성 필드와 함께 문서화된 추가 특수 키를 지원합니다. 모든 비밀 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 조합의 불일치를 확인하지 않습니다.

TLS(Transport Layer Security)

TLS URL 사용('http://...' or 'ssl://...') 보안을 사용하지 않으면 기본 TLS 서버 측 인증을 사용할 수 있습니다. 시크릿을 포함한 추가 TLS 기능은 다음 선택적 필드를 설정하여 활성화합니다.

- TLS.crt: (문자열) 클라이언트 인증서를 포함하는 파일 이름입니다. 상호 인증을 활성화합니다. tls.key 가 필요합니다.
- TLS.key: (문자열) 클라이언트 인증서의 잠금을 해제하는 개인 키가 포함된 파일 이름입니다. tls.crt 가 필요합니다.
- 암호: (문자열) 인코딩된 TLS 개인 키를 디코딩하는 Passphrase입니다. tls.key 가 필요합니다.
- ca-bundle.crt: 서버 인증을 위해 고객 CA의 파일 이름.

사용자 이름 및 암호

- 사용자 이름: (문자열) 인증 사용자 이름. 암호 가 필요합니다.
- 암호: (문자열) 인증 암호. 사용자 이름 필요 .

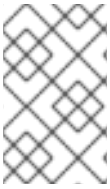
SASL(Simple Authentication Security Layer)

- SASL.enable (boolean) Explicitly enable 또는 disable SASL. 누락된 경우 다른 sasl. 키가 설정되면 SASL이 자동으로 활성화됩니다.
- SASL.mechanisms: 허용되는 SASL 메커니즘 이름 목록. 누락되거나 비어 있는 경우 시스템 기본값이 사용됩니다.
- SASL.allow-insecure: (boolean) 일반 텍스트 암호를 보내는 메커니즘을 허용합니다. 기본값은 false입니다.

7.1.1. 보안 생성

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

```
$ oc create secret generic -n openshift-logging <my-secret> \
 --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

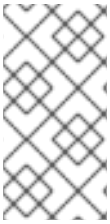
최상의 결과를 위해 일반 또는 불투명 보안을 사용하는 것이 좋습니다.

7.2. OPENSHIFT LOGGING 5.1에서 지원되는 로그 데이터 출력 유형

Red Hat OpenShift Logging 5.1은 로그 데이터를 대상 로그 수집기로 전송하는 데 필요한 다음과 같은 출력 유형 및 프로토콜을 제공합니다.

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

출력 대상	프로토콜	테스트에 사용
elasticsearch	elasticsearch	Elasticsearch 6.8.1 Elasticsearch 6.8.4 Elasticsearch 7.12.2
fluentdForward	fluentd forward v1	fluentd 1.7.4 logstash 7.10.1
kafka	kafka 0.11	kafka 2.4.1 kafka 2.7.0
syslog	RFC-3164, RFC-5424	rsyslog-8.39.0



참고

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

7.3. OPENSHIFT LOGGING 5.2에서 지원되는 로그 데이터 출력 유형

Red Hat OpenShift Logging 5.2는 로그 데이터를 대상 로그 수집기로 전송하는 데 필요한 다음과 같은 출력 유형 및 프로토콜을 제공합니다.

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

출력 대상	프로토콜	테스트에 사용
Amazon CloudWatch	HTTPS를 통한 REST	Amazon CloudWatch의 현재 버전
elasticsearch	elasticsearch	Elasticsearch 6.8.1 Elasticsearch 6.8.4 Elasticsearch 7.12.2
fluentdForward	fluentd forward v1	fluentd 1.7.4 logstash 7.10.1
Loki	HTTP 및 HTTPS를 통한 REST	OCP and Grafana labs에 배포된 Loki 2.3.0
kafka	kafka 0.11	kafka 2.4.1 kafka 2.7.0
syslog	RFC-3164, RFC-5424	rsyslog-8.39.0

7.4. OPENSHIFT LOGGING 5.3에서 지원되는 로그 데이터 출력 유형

Red Hat OpenShift Logging 5.3은 로그 데이터를 대상 로그 수집기로 전송하기 위해 다음과 같은 출력 유형 및 프로토콜을 제공합니다.

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

출력 대상	프로토콜	테스트에 사용
Amazon CloudWatch	HTTPS를 통한 REST	Amazon CloudWatch의 현재 버전
elasticsearch	elasticsearch	Elasticsearch 7.10.1

출력 대상	프로토콜	테스트에 사용
fluentdForward	fluentd forward v1	fluentd 1.7.4 logstash 7.10.1
Loki	HTTP 및 HTTPS를 통한 REST	OCP에 배포된 2.2.1
kafka	kafka 0.11	kafka 2.7.0
syslog	RFC-3164, RFC-5424	rsyslog-8.39.0

7.5. OPENSHIFT LOGGING 5.4에서 지원되는 로그 데이터 출력 유형

Red Hat OpenShift Logging 5.4는 로그 데이터를 대상 로그 수집기로 전송하기 위해 다음과 같은 출력 유형 및 프로토콜을 제공합니다.

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

출력 대상	프로토콜	테스트에 사용
Amazon CloudWatch	HTTPS를 통한 REST	Amazon CloudWatch의 현재 버전
elasticsearch	elasticsearch	Elasticsearch 7.10.1
fluentdForward	fluentd forward v1	fluentd 1.14.5 logstash 7.10.1
Loki	HTTP 및 HTTPS를 통한 REST	OCP에 배포된 2.2.1
kafka	kafka 0.11	kafka 2.7.0
syslog	RFC-3164, RFC-5424	rsyslog-8.39.0

7.6. OPENSHIFT LOGGING 5.5에서 지원되는 로그 데이터 출력 유형

Red Hat OpenShift Logging 5.5는 로그 데이터를 대상 로그 수집기로 전송하기 위해 다음과 같은 출력 유형 및 프로토콜을 제공합니다.

Red Hat은 다음 표에 표시된 각 조합을 테스트합니다. 그러나 이러한 프로토콜을 수집하는 광범위한

범위 대상 로그 수집기로 로그 데이터를 전송할 수 있어야 합니다.

출력 대상	프로토콜	테스트에 사용
Amazon CloudWatch	HTTPS를 통한 REST	Amazon CloudWatch의 현재 버전
elasticsearch	elasticsearch	Elasticsearch 7.10.1
fluentdForward	fluentd forward v1	fluentd 1.14.6 logstash 7.10.1
Loki	HTTP 및 HTTPS를 통한 REST	OCP에 배포된 2.5.0
kafka	kafka 0.11	kafka 2.7.0
syslog	RFC-3164, RFC-5424	rsyslog-8.39.0

7.7. OPENSHIFT LOGGING 5.6에서 지원되는 로그 데이터 출력 유형

Red Hat OpenShift Logging 5.6은 로그 데이터를 대상 로그 수집기로 전송하기 위해 다음과 같은 출력 유형 및 프로토콜을 제공합니다.

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

출력 대상	프로토콜	테스트에 사용
Amazon CloudWatch	HTTPS를 통한 REST	Amazon CloudWatch의 현재 버전
elasticsearch	elasticsearch	Elasticsearch 6.8.23 Elasticsearch 7.10.1 Elasticsearch 8.6.1
fluentdForward	fluentd forward v1	fluentd 1.14.6 logstash 7.10.1
Loki	HTTP 및 HTTPS를 통한 REST	OCP에 배포된 2.5.0
kafka	kafka 0.11	kafka 2.7.0

출력 대상	프로토콜	테스트에 사용
syslog	RFC-3164, RFC-5424	rsyslog-8.39.0

중요

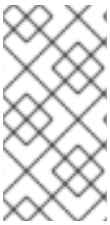
Fluentd는 **Elasticsearch 8**을 **5.6.2**로 지원하지 않습니다. **vector**는 **5.7.0** 이전의 **fluentd/logstash/ECDHE**를 지원하지 않습니다.

7.8. 외부 ELASTICSEARCH 인스턴스로 로그 전달

선택적으로 내부 **OpenShift Container Platform Elasticsearch** 인스턴스에 추가하거나 대신 외부 **Elasticsearch** 인스턴스에 로그를 전달할 수 있습니다. **OpenShift Container Platform**에서 로그 데이터를 수신하도록 외부 로그 집계기를 구성해야 합니다.

외부 **Elasticsearch** 인스턴스에 대한 로그 전달을 구성하려면 해당 인스턴스에 대한 출력과 출력을 사용하는 파이프라인이 있는 **ClusterLogForwarder** 사용자 정의 리소스(**CR**)를 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. **default** 출력을 생성할 필요가 없습니다. **default** 출력을 구성하는 경우 **default** 출력이 **Red Hat OpenShift Logging Operator**용으로 예약되므로 오류 메시지가 나타납니다.



참고

내부 **OpenShift Container Platform Elasticsearch** 인스턴스에만 로그를 전달하려는 경우 **ClusterLogForwarder CR**을 생성할 필요가 없습니다.

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: instance (1)
  namespace: openshift-logging (2)
spec:
  outputs:
   - name: elasticsearch-insecure (3)
     type: "elasticsearch" (4)
     url: http://elasticsearch.insecure.com:9200 (5)
   - name: elasticsearch-secure
     type: "elasticsearch"
     url: https://elasticsearch.secure.com:9200 (6)
     secret:
       name: es-secret (7)
  pipelines:
   - name: application-logs (8)
     inputRefs: (9)
     - application
     - audit
     outputRefs:
     - elasticsearch-secure (10)
     - default (11)
     parse: json (12)
     labels:
       myLabel: "myValue" (13)
   - name: infrastructure-audit-logs (14)
     inputRefs:
     - infrastructure
     outputRefs:
     - elasticsearch-insecure
     labels:
       logs: "audit-infra"
```

(1) ClusterLogForwarder CR의 이름은 instance여야 합니다.

(2) ClusterLogForwarder CR의 네임스페이스는 openshift-logging이어야 합니다.

(3)

(4) elasticsearch 유형을 지정합니다.

(5)

6

보안 연결의 경우 **secret**을 지정하여 인증하는 **https** 또는 **http URL**을 지정할 수 있습니다.

7

https 접두사의 경우 **TLS** 통신의 엔드포인트에 필요한 보안 이름을 지정합니다. 시크릿은 **openshift-logging** 프로젝트에 있어야 하며 해당하는 각 인증서를 가리키는 **tls.crt,tls.key** 및 **ca-bundle.crt** 키가 있어야 합니다. 그러지 않으면 **http** 및 **https** 접두사의 경우 사용자 이름과 암호가 포함된 시크릿을 지정할 수 있습니다. 자세한 내용은 다음 "예: 사용자 이름과 암호가 포함된 시크릿 설정".

8

선택 사항: 파이프라인의 이름을 지정합니다.

9

파이프라인을 사용하여 전달할 로그 유형 (**application, infrastructure,** 또는 **audit**)을 지정합니다.

10

이 파이프라인으로 로그를 전달할 때 사용할 출력 이름을 지정합니다.

11

선택 사항: 로그를 내부 **Elasticsearch** 인스턴스로 보내려면 기본 출력을 지정합니다.

12

선택 사항: 구조화된 필드에서 **JSON** 오브젝트로 구조화된 **JSON** 로그 항목을 전달할지 여부를 지정합니다. 로그 항목에 유효한 구조화된 **JSON**이 포함되어야 합니다. 그렇지 않으면 **OpenShift Logging**이 **structured** 필드를 제거하고 대신 기본 인덱스인 **app-00000x**로 로그 항목을 보냅니다.

13

선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

14

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

- 파이프라인을 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
```

예제: 사용자 이름 및 암호가 포함된 보안 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: dGVzdHVzZXJuYW1lCg==
  password: dGVzdHBhc3N3b3JkCg==
```

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.

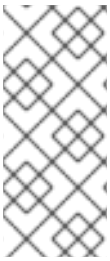
4. CR 오브젝트를 생성합니다.

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

7.9. 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.



참고

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

사전 요구 사항

-

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

절차

1. 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
     parse: json 11
     labels:
       clusterId: "C1234" 12
   - name: forward-to-fluentd-insecure 13
     inputRefs:
     - infrastructure
     outputRefs:
     - fluentd-server-insecure
     labels:
       clusterId: "C1234"
```

1 ClusterLogForwarder CR의 이름은 instance여야 합니다.

2 ClusterLogForwarder CR의 네임스페이스는 openshift-logging이어야 합니다.

3

출력 이름을 지정합니다.

4

fluentdForward 유형을 지정합니다.

5

유효한 절대 **URL**로 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 **FQDN**이어야 합니다.

6

tls 접두사를 사용하는 경우 **TLS** 통신을 위해 끝점에서 요구하는 시크릿 이름을 지정해야 합니다. 시크릿은 **openshift-logging** 프로젝트에 있어야 하며 해당하는 각 인증서를 가리키는 **tls.crt,tls.key** 및 **ca-bundle.crt** 키가 있어야 합니다. 그렇지 않으면 **http** 및 **https** 접두사의 경우 사용자 이름 및 암호가 포함된 보안을 지정할 수 있습니다. 자세한 내용은 다음 **"예: 사용자 이름과 암호가 포함된 시크릿 설정"**.

7

선택 사항: 파이프라인의 이름을 지정합니다.

8

파이프라인을 사용하여 전달할 로그 유형 (**application, infrastructure,** 또는 **audit**)을 지정합니다.

9

이 파이프라인으로 로그를 전달할 때 사용할 출력 이름을 지정합니다.

10

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

11

선택 사항: 구조화된 필드에서 **JSON** 오브젝트로 구조화된 **JSON** 로그 항목을 전달할지 여부를 지정합니다. 로그 항목에 유효한 구조화된 **JSON**이 포함되어야 합니다. 그렇지 않으면 **OpenShift Logging**이 **structured** 필드를 제거하고 대신 기본 인덱스인 **app-00000x**로 로그 항목을 보냅니다.

12

13

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

- 파이프라인을 설명하는 이름입니다.
- inputRefs는 pipeline: application, infrastructure 또는 audit를 사용하여 전달할 로그 유형입니다.
- outputRefs는 사용할 출력의 이름입니다.
- 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

2. CR 오브젝트를 생성합니다.

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

7.9.1. Logstash가 fluentd에서 데이터를 수집할 수 있도록 nanosecond precision 사용

fluentd에서 로그 데이터를 수집하려면 Logstash 구성 파일에서 nanosecond precision을 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 } }
```

7.10. SYSLOG 프로토콜을 사용하여 로그 전달

syslog RFC3164 또는 RFC5424 프로토콜을 사용하여 기본 **Elasticsearch** 로그 저장소 대신 또는 기본 **Elasticsearch** 로그 저장소에 더하여 해당 프로토콜을 수락하도록 구성된 외부 로그 집계기에 로그 사본을 보낼 수 있습니다. **OpenShift Container Platform**에서 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.

참고

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

사전 요구 사항

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

절차

1. **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: rsyslog-east 3
     type: syslog 4
     syslog: 5
       facility: local0
       rfc: RFC3164
       payloadKey: message
       severity: informational
```

```
    url: 'tls://rsyslogserver.east.example.com:514' 6
    secret: 7
      name: syslog-secret
  - name: rsyslog-west
    type: syslog
    syslog:
     appName: myapp
     facility: user
     msgID: mymsg
     procID: myproc
     rfc: RFC5424
     severity: debug
    url: 'udp://rsyslogserver.west.example.com:514'
pipelines:
  - name: syslog-east 8
    inputRefs: 9
    - audit
    - application
    outputRefs: 10
    - rsyslog-east
    - default 11
    parse: json 12
    labels:
      secure: "true" 13
      syslog: "east"
  - name: syslog-west 14
    inputRefs:
    - infrastructure
    outputRefs:
    - rsyslog-west
    - default
    labels:
      syslog: "west"
```

1

ClusterLogForwarder CR의 이름은 instance여야 합니다.

2

3

출력 이름을 지정합니다.

4

syslog 유형을 지정합니다.

5

6

외부 **syslog** 인스턴스의 **URL** 및 포트를 지정합니다. **udp**(비보안), **tcp**(비보안) 또는 **tls**(보안 **TCP**) 프로토콜을 사용할 수 있습니다. **CIDR** 주석을 사용하는 클러스터 전체 프록시가 활성화된 경우 출력은 **IP** 주소가 아닌 서버 이름 또는 **FQDN**이어야 합니다.

7

tls 접두사를 사용하는 경우 **TLS** 통신을 위해 끝점에서 요구하는 시크릿 이름을 지정해야 합니다. 시크릿은 **openshift-logging** 프로젝트에 있어야 하며 해당하는 각 인증서를 가리키는 **tls.crt**,**tls.key** 및 **ca-bundle.crt** 키가 있어야 합니다.

8

선택 사항: 파이프라인의 이름을 지정합니다.

9

파이프라인을 사용하여 전달할 로그 유형 (**application**, **infrastructure**, 또는 **audit**)을 지정합니다.

10

이 파이프라인으로 로그를 전달할 때 사용할 출력 이름을 지정합니다.

11

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

12

선택 사항: 구조화된 필드에서 **JSON** 오브젝트로 구조화된 **JSON** 로그 항목을 전달할지 여부를 지정합니다. 로그 항목에 유효한 구조화된 **JSON**이 포함되어야 합니다. 그렇지 않으면 **OpenShift Logging**이 **structured** 필드를 제거하고 대신 기본 인덱스인 **app-00000x**로 로그 항목을 보냅니다.

13

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

14

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

- 파이프라인을 설명하는 이름입니다.
- inputRefs는 pipeline: application, infrastructure 또는 audit를 사용하여 전달할 로그 유형입니다.
- outputRefs는 사용할 출력의 이름입니다.
- 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

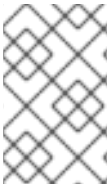
2. CR 오브젝트를 생성합니다.

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

7.10.1. 메시지 출력에 로그 소스 정보 추가

AddLogSource 필드를 ClusterLogForwarder CR(사용자 정의 리소스)에 추가하여 namespace_name,pod_name, container_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 둘 다와 호환됩니다.

AddLogSource없는 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}
```

AddLogSource를 사용한 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}
```

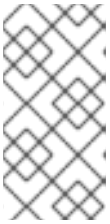
7.10.2. Syslog 매개변수

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

- 기능: syslog 기능. 값은 10진수 정수 또는 대소문자를 구분하지 않는 키워드일 수 있습니다.
  - 커널 메시지의 경우 0 또는 kern
  - 사용자 수준 메시지의 경우 1 또는 user, 기본값입니다.
  - 2 또는 mail 시스템용 메일

- 시스템 데몬의 경우 **3** 또는 **daemon**
- 보안/인증 메시지의 경우 **4** 또는 **auth**
- **syslogd**에 의해 내부적으로 생성된 메시지의 경우 **5** 또는 **syslog**
- 라인 프린터 하위 시스템의 경우 **6** 또는 **lpr**
- 네트워크 뉴스 서브 시스템의 경우 **7** 또는 **news**
- **UUCP** 하위 시스템의 경우 **8** 또는 **uucp**
- 시계 데몬의 경우 **9** 또는 **cron**
- 보안 인증 메시지의 경우 **10** 또는 **authpriv**
- **FTP** 데몬의 경우 **11** 또는 **ftp**
- **NTP** 하위 시스템의 경우 **12** 또는 **ntp**
- **syslog** 감사 로그의 경우 **13** 또는 **security**
- **syslog** 경고 로그의 경우 **14** 또는 **console**
- 스케줄링 데몬의 경우 **15** 또는 **solaris-cron**
- 로컬에서 사용되는 시설의 경우 **16 – 23** 또는 **local0 – local7**

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



참고

payloadKey 매개변수를 구성하면 다른 매개 변수가 syslog로 전달되지 않습니다.

- rfc: syslog를 사용하여 로그를 전송하는 데 사용할 RFC입니다. 기본값은 RFC5424입니다.

- 심각도: 발신 syslog 레코드에 설정할 syslog 심각도 입니다. 값은 10진수 정수 또는 대소문자를 구분하지 않는 키워드일 수 있습니다.
  - 시스템을 사용할 수 없음을 나타내는 메시지의 경우 0 또는 Emergency
  - 조치를 즉시 취해야 함을 나타내는 메시지의 경우 1 또는 Alert
  - 위험 상태를 나타내는 메시지의 경우 2 또는 Critical
  - 오류 상태를 나타내는 메시지의 경우 3 또는 Error
  - 경고 조건을 나타내는 메시지의 경우 4 또는 Warning
  - 정상이지만 중요한 조건을 나타내는 메시지의 경우 5 또는 Notice
  - 정보성 메시지를 나타내는 메시지의 경우 6 또는 Informational
  - 디버그 수준 메시지를 나타내는 메시지의 경우 7 또는 Debug, 기본값

- 태그: tag는 syslog 메시지에서 태그로 사용할 레코드 필드를 지정합니다.

- **trimPrefix:** 태그에서 지정된 접두사를 제거합니다.

7.10.3. 추가 RFC5424 syslog 매개변수

RFC5424에는 다음 매개변수가 적용됩니다.

- **appName: APP-NAME**은 로그를 전송한 애플리케이션을 식별하는 자유 텍스트 문자열입니다. **RFC5424**에 대해 지정해야 합니다.

- **msgID: MSGID**는 메시지 유형을 식별하는 자유 텍스트 문자열입니다. **RFC5424**에 대해 지정해야 합니다.

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

7.11. AMAZON CLOUDVIEW로 로그 전달

AWS(Amazon Web Services)에서 호스팅하는 모니터링 및 로그 스토리지 서비스인 Amazon CloudMonitor에 로그를 전달할 수 있습니다. 기본 OpenShift Logging 관리 Elasticsearch 로그 저장소에 추가하거나 대신 CloudMonitor에 로그를 전달할 수 있습니다.

CloudMonitor에 대한 로그 전달을 구성하려면 CloudMonitor의 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 base64로 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: QUtJQUlPU0ZPRE5ON0VYQU1QTEUK
  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: instance 1
  namespace: openshift-logging 2
spec:
  outputs:
   - name: cw 3
     type: cloudwatch 4
     cloudwatch:
       groupBy: logType 5
       groupPrefix: <group prefix> 6
       region: us-east-2 7
     secret:
        name: cw-secret 8
  pipelines:
    - name: infra-logs 9
      inputRefs: 10
        - infrastructure
        - audit
        - application
      outputRefs:
        - cw 11
```

1. ClusterLogForwarder CR의 이름은 instance여야 합니다.

2.

3. 출력 이름을 지정합니다.

4. cloudwatch 유형을 지정합니다.

5.

- **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 **UUID**에 대한 새 로그 그룹을 생성합니다. 인프라 및 감사 로그를 위해 별도의 로그 그룹도 생성합니다.

6 선택 사항: 로그 그룹 이름에 기본 **infrastructureName** 접두사를 대체할 문자열을 지정합니다.

7 **AWS** 리전을 지정합니다.

8 **AWS** 인증 정보가 포함된 시크릿의 이름을 지정합니다.

9 선택 사항: 파이프라인의 이름을 지정합니다.

10

11 이 파이프라인으로 로그를 전달할 때 사용할 출력 이름을 지정합니다.

4. **CR** 오브젝트를 생성합니다.

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

예제: Amazon CloudMonitor와 함께 ClusterLogForwarder 사용

여기에 **ClusterLogForwarder** 사용자 정의 리소스(CR)의 예와 **Amazon CloudMonitor**로 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 app 이라는 네임스페이스에서 busybox Pod를 실행합니다. busybox 포드는 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 실행되는 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 파이프라인에 대한 입력으로 구성합니다. 또한 이 파이프라인을 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
      },
...
```

예제: 로그 그룹 이름에 접두사 사용자 정의

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"
```

예제: 애플리케이션 네임스페이스 이름 뒤에 로그 그룹 이름 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 **UUID**에 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.

예제: 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. 그렇지 않으면 앞의 "예: 대신 애플리케이션 네임스페이스 이름에 대한 로그 그룹 이름 지정" 섹션.

애플리케이션 네임스페이스 UUID 후에 로그 그룹의 이름을 지정하려면 ClusterLogForwarder CR에서 groupBy 필드의 값을 namespaceUUID로 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.

7.12. LOKI로 로그 전달

내부 기본 OpenShift Container Platform Elasticsearch 인스턴스 대신 외부 Loki 로깅 시스템으로 로그를 전달할 수 있습니다.

Loki에 대한 로그 전달을 구성하려면 Loki에 대한 출력과 출력을 사용하는 파이프라인이 있는

ClusterLogForwarder 사용자 정의 리소스(CR)를 생성해야 합니다. Loki의 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: instance 1
  namespace: openshift-logging 2
spec:
  outputs:
   - name: loki-insecure 3
     type: "loki" 4
     url: http://loki.insecure.com:3100 5
   - name: loki-secure
     type: "loki"
     url: https://loki.secure.com:3100 6
     secret:
       name: loki-secret 7
  pipelines:
   - name: application-logs 8
     inputRefs: 9
     - application
     - audit
     outputRefs:
     - loki-secure 10
     loki:
       tenantKey: kubernetes.namespace_name 11
       labelKeys: kubernetes.labels.foo 12
```

1. ClusterLogForwarder CR의 이름은 instance여야 합니다.

2. ClusterLogForwarder CR의 네임스페이스는 openshift-logging이어야 합니다.

3.

4

유형을 "loki"로 지정합니다.

5

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.

6

보안 연결의 경우 secret을 지정하여 인증하는 https 또는 http URL을 지정할 수 있습니다.

7

https 접두사의 경우 TLS 통신의 엔드포인트에 필요한 보안 이름을 지정합니다. 시크릿은 openshift-logging 프로젝트에 있어야 하며 해당하는 각 인증서를 가리키는 tls.crt,tls.key 및 ca-bundle.crt 키가 있어야 합니다. 그러지 않으면 http 및 https 접두사의 경우 사용자 이름과 암호가 포함된 시크릿을 지정할 수 있습니다. 자세한 내용은 다음 "예: 사용자 이름과 암호가 포함된 시크릿 설정".

8

선택 사항: 파이프라인의 이름을 지정합니다.

9

10

이 파이프라인으로 로그를 전달할 때 사용할 출력 이름을 지정합니다.

11

선택 사항: Loki의 TenantID 필드 값을 생성할 meta-data 키 필드를 지정합니다. 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.

12

선택 사항: 기본 Loki 레이블을 바꾸려면 meta-data 필드 키 목록을 지정합니다. Loki 레이블 이름은 정규식 [a-zA-Z_:][a-zA-Z0-9_:]*와 일치해야 합니다. 메타 데이터 키의 잘못된 문자는 레이블 이름을 형성하기 위해 _로 대체됩니다. 예를 들어 kubernetes.labels.foo 메타 데이터 키는 Loki 레이블 kubernetes_labels_foo가 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_config를 참조하십시오. 쿼리 필터를 사용하여 로그 레코드 필드를 기반으로 쿼리할 수 있습니다.

참고

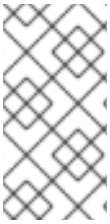
Loki는 타임스탬프에 의해 로그 스트림을 올바르게 정렬해야 하므로 labelKeys에는 항상 kubernetes_host 레이블 세트가 포함됩니다. 이렇게 하면 각 스트림이 단일 호스트에서 시작되도록 하여 서로 다른 호스트의 클록 차이로 인해 타임스탬프가 무질서해지는 것을 방지할 수 있습니다.

2. CR 오브젝트를 생성합니다.

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

7.12.1. Loki "주문 부족" 오류 문제 해결

Fluentd가 대량의 메시지 블록을 속도 제한을 초과하는 Loki 로깅 시스템에 전달하는 경우 Loki는 "주문 부족" 오류를 생성합니다. 이 문제를 해결하려면 Loki 서버 구성 파일인 loki.yaml에서 일부 값을 업데이트합니다.



참고

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

조건

- ClusterLogForwarder 사용자 정의 리소스는 로그를 Loki로 전달하도록 구성되어 있습니다.
- 시스템에서 2MB보다 큰 메시지 블록을 Loki에 보냅니다. 예를 들면 다음과 같습니다.

```
"values":[["1630410392689800468","{\"kind\":\"Event\",\"apiVersion\":\
.......
......
......
......
\"received_at\":\"2021-08-31T11:46:32.800278+00:00\",\"version\":\"1.7.4
1.6.0\"}},\"@timestamp\":\"2021-08-
```

> 31T11:46:32.799692+00:00\",\"viaq_index_name\":\"audit-write\",\"viaq_msg_id\":\"MzFjYjJkZjItNjY0MC00YWU4LWIwMTEtNGNmM2E5ZmViMGU4\",\"log_type\":\"audit\"}"]]}]}

- **oc logs -c fluentd**를 입력하면 **OpenShift Logging** 클러스터의 **Fluentd** 로그에 다음 메시지가 표시됩니다.

> **429 Too Many Requests Ingestion rate limit exceeded (limit: 8388608 bytes/sec) while attempting to ingest '2140' lines totaling '3285284' bytes**
>
> **429 Too Many Requests Ingestion rate limit exceeded' or '500 Internal Server Error rpc error: code = ResourceExhausted desc = grpc: received message larger than max (5277702 vs. 4194304)'**

- **Loki** 서버에서 로그를 열면 다음과 같은 **entry out of order** 메시지가 표시됩니다.

> **,\nentry with timestamp 2021-08-18 05:58:55.061936 +0000 UTC ignored, reason: 'entry out of order' for stream:**
>
> **{fluentd_thread=\"flush_thread_0\", log_type=\"audit\"},\nentry with timestamp 2021-08-18 06:01:18.290229 +0000 UTC ignored, reason: 'entry out of order' for stream: {fluentd_thread="flush_thread_0", log_type="audit"}**

절차

1. **Loki** 서버의 **loki.yaml** 구성 파일에서 다음 필드를 여기에 표시된 값으로 업데이트합니다.
   - **grpc_server_max_recv_msg_size: 8388608**
   - **chunk_target_size: 8388608**
   - **ingestion_rate_mb: 8**
   - **ingestion_burst_size_mb: 16**
2. **Loki .yaml**의 변경 사항을 **Loki** 서버에 적용합니다.

loki.yaml 파일 예

```
auth_enabled: false

server:
  http_listen_port: 3100
  grpc_listen_port: 9096
  grpc_server_max_recv_msg_size: 8388608

ingester:
  wal:
    enabled: true
    dir: /tmp/wal
  lifecycler:
    address: 127.0.0.1
    ring:
      kvstore:
        store: inmemory
      replication_factor: 1
    final_sleep: 0s
  chunk_idle_period: 1h       # Any chunk not receiving new logs in this time will be flushed
  chunk_target_size: 8388608
  max_chunk_age: 1h           # All chunks will be flushed when they hit this age, default is 1h
  chunk_retain_period: 30s    # Must be greater than index read cache TTL if using an index
cache (Default index read cache TTL is 5m)
  max_transfer_retries: 0     # Chunk transfers disabled

schema_config:
  configs:
    - from: 2020-10-24
      store: boltdb-shipper
      object_store: filesystem
      schema: v11
      index:
        prefix: index_
        period: 24h

storage_config:
  boltdb_shipper:
    active_index_directory: /tmp/loki/boltdb-shipper-active
    cache_location: /tmp/loki/boltdb-shipper-cache
    cache_ttl: 24h         # Can be increased for faster performance over longer query periods,
uses more disk space
    shared_store: filesystem
  filesystem:
    directory: /tmp/loki/chunks

compactor:
  working_directory: /tmp/loki/boltdb-shipper-compactor
  shared_store: filesystem

limits_config:
  reject_old_samples: true
  reject_old_samples_max_age: 12h
  ingestion_rate_mb: 8
```

```
  ingestion_burst_size_mb: 16

chunk_store_config:
  max_look_back_period: 0s

table_manager:
  retention_deletes_enabled: false
  retention_period: 0s

ruler:
  storage:
    type: local
    local:
      directory: /tmp/loki/rules
  rule_path: /tmp/loki/rules-temp
  alertmanager_url: http://localhost:9093
  ring:
    kvstore:
      store: inmemory
  enable_api: true
```

추가 리소스

- Loki 구성

추가 리소스

- 로그 레코드 필드.
- Loki 서버 구성

7.13. 특정 프로젝트의 애플리케이션 로그 전달

Cluster Log Forwarder를 사용하여 특정 프로젝트의 애플리케이션 로그 사본을 외부 로그 수집기로 보낼 수 있습니다. 기본 **Elasticsearch** 로그 저장소를 사용하여 추가하거나 대신 이 작업을 수행할 수 있습니다. **OpenShift Container 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

1. **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'
  inputs: 7
   - name: my-app-logs
     application:
       namespaces:
       - my-project
  pipelines:
   - name: forward-to-fluentd-insecure 8
     inputRefs: 9
     - my-app-logs
     outputRefs: 10
     - fluentd-server-insecure
     parse: json 11
     labels:
       project: "my-project" 12
   - name: forward-to-fluentd-secure 13
     inputRefs:
     - application
     - audit
     - infrastructure
     outputRefs:
     - fluentd-server-secure
     - default
     labels:
       clusterId: "C1234"
```

1

2

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

3

출력 이름을 지정합니다.

4

출력 유형을 **elasticsearch, fluentdForward, syslog** 또는 **kafka**로 지정합니다.

5

외부 로그 집계기의 **URL** 및 포트를 유효한 절대 **URL**로 지정합니다. **CIDR** 주석을 사용하는 클러스터 전체 프록시가 활성화된 경우 출력은 **IP** 주소가 아닌 서버 이름 또는 **FQDN**이어야 합니다.

6

tls 접두사를 사용하는 경우 **TLS** 통신을 위해 끝점에서 요구하는 시크릿 이름을 지정해야 합니다. 시크릿은 **openshift-logging** 프로젝트에 있어야 하며 각각의 인증서를 가리키는 **tls.crt, tls.key,** 및 **ca-bundle.crt** 키가 있어야 합니다.

7

지정된 프로젝트에서 애플리케이션 로그를 필터링하기 위한 입력 구성입니다.

8

입력을 사용하여 프로젝트 애플리케이션 로그를 외부 **Fluentd** 인스턴스로 보내는 파이프라인 구성입니다.

9

my-app-logs 입력입니다.

10

사용할 출력의 이름입니다.

11

선택 사항: 구조화된 필드에서 **JSON** 오브젝트로 구조화된 **JSON** 로그 항목을 전달할지 여부를 지정합니다. 로그 항목에 유효한 구조화된 **JSON**이 포함되어야 합니다. 그렇지 않으면 **OpenShift Logging**이 **structured** 필드를 제거하고 대신 기본 인덱스인 **app-**

00000x로 로그 항목을 보냅니다.

12

선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

13

로그를 다른 로그 집계기로 보내는 파이프라인 구성입니다.

- 선택 사항: 파이프라인의 이름을 지정합니다.
- 파이프라인을 사용하여 전달할 로그 유형 (application, infrastructure, 또는 audit)을 지정합니다.
- 이 파이프라인으로 로그를 전달할 때 사용할 출력 이름을 지정합니다.
- 선택 사항: 로그를 내부 Elasticsearch 인스턴스로 전달하려면 default 출력을 지정합니다.
- 선택 사항: 문자열. 로그에 추가할 하나 이상의 레이블입니다.

2. CR 오브젝트를 생성합니다.

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

7.14. 특정 POD에서 애플리케이션 로그 전달

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

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

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

절차

1. ClusterLogForwarder CR 오브젝트를 정의하는 YAML 파일을 생성하거나 편집합니다. 파일에서 다음 예와 같이 inputs[].name.application.selector.matchLabels에서 간단한 동일성 기반 선택기를 사용하여 Pod 레이블을 지정합니다.

ClusterLogForwarder CR YAML 파일의 예

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: instance 1
  namespace: openshift-logging 2
spec:
  pipelines:
    - inputRefs: [ myAppLogData ] 3
      outputRefs: [ default ] 4
      parse: json 5
  inputs: 6
    - name: myAppLogData
      application:
        selector:
          matchLabels: 7
            environment: production
            app: nginx
        namespaces: 8
        - app1
        - app2
  outputs: 9
    - default
    ...
```

1 ClusterLogForwarder CR의 이름은 instance여야 합니다.

2 ClusterLogForwarder CR의 네임스페이스는 openshift-logging이어야 합니다.

3

inputs[].name에서 하나 이상의 쉼표로 구분된 값을 지정합니다.

4

outputs[]에서 하나 이상의 쉼표로 구분된 값을 지정합니다.

5

선택 사항: 구조화된 필드에서 **JSON** 오브젝트로 구조화된 **JSON** 로그 항목을 전달할지 여부를 지정합니다. 로그 항목에 유효한 구조화된 **JSON**이 포함되어야 합니다. 그렇지 않으면 **OpenShift Logging**이 **structured** 필드를 제거하고 대신 기본 인덱스인 **app-00000x**로 로그 항목을 보냅니다.

6

고유한 **pod** 레이블 집합이 있는 각 애플리케이션에 대해 고유한 **inputs[].name**을 정의합니다.

7

수집하려는 로그 데이터가 있는 **Pod** 라벨의 키-값 쌍을 지정합니다. 키뿐만 아니라 키와 값 모두를 지정해야 합니다. 선택하려면 **Pod**가 모든 키-값 쌍과 일치해야 합니다.

8

선택 사항: 하나 이상의 네임스페이스를 지정합니다.

9

로그 데이터를 전달할 출력을 하나 이상 지정합니다. 여기에 표시된 **default** 출력 (선택 사항)은 로그 데이터를 내부 **Elasticsearch** 인스턴스로 전송합니다.

2. 선택 사항: 로그 데이터 수집을 특정 네임스페이스로 제한하려면 위 예제와 같이 **inputs[].name.application.namespaces** 를 사용합니다.

3. 선택 사항: 다른 **Pod** 라벨이 있는 추가 애플리케이션에서 동일한 파이프라인으로 로그 데이터를 보낼 수 있습니다.

   a. **Pod** 레이블의 고유한 조합마다 표시된 항목과 유사한 추가 **inputs[].name** 섹션을 생성합니다.

b. 이 애플리케이션의 **Pod** 레이블과 일치하도록 **selectors**를 업데이트합니다.

c. 새 **inputs[].name** 값을 **inputRefs**에 추가합니다. 예를 들면 다음과 같습니다.

```
- inputRefs: [ myAppLogData, myOtherAppLogData ]
```

4. **CR** 오브젝트를 생성합니다.

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

추가 리소스

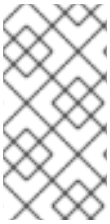
- **Kubernetes**의 **matchLabels**에 대한 자세한 내용은 세트 기반 요구 사항을 지원하는 리소스를 참조하십시오.

7.15. OVN 네트워크 정책 감사 로그 수집

OVN-Kubernetes Pod의 **/var/log/ovn/acl-audit-log.log** 파일에서 **OVN** 네트워크 정책 감사 로그를 수집하여 로깅 서버로 전달할 수 있습니다.

사전 요구 사항

- **OpenShift Container Platform** 버전 **4.8** 이상을 사용하고 있어야 합니다.
- 클러스터 로깅 **5.2** 이상을 사용하고 있어야 합니다.
- **ClusterLogForwarder CR**(사용자 정의 리소스) 오브젝트가 이미 설정되어 있습니다.
- **OpenShift Container Platform** 클러스터는 **OVN-Kubernetes** 네트워크 정책 감사 로깅을 위해 구성되어 있습니다. 다음 "추가 리소스" 섹션을 참조하십시오.



참고

감사 데이터를 저장하는 로깅 서버는 규정 준수 및 보안을 위한 조직 및 정부 요구 사항을 충족해야 합니다.

절차

1. 타사 시스템으로 로그를 전달하는 방법에 대한 다른 항목에 설명된 대로 **ClusterLogForwarder CR** 오브젝트를 정의하는 **YAML** 파일을 생성하거나 편집합니다.

2. **YAML** 파일에서 파이프라인의 **inputRefs** 요소에 **audit** 로그 유형을 추가합니다. 예를 들면 다음과 같습니다.

```
pipelines:
 - name: audit-logs
   inputRefs:
    - audit (1)
   outputRefs:
    - secure-logging-server (2)
```

(1) 입력할 로그 유형 중 하나로 **audit**을 지정합니다.

(2) 로깅 서버에 연결하는 출력을 지정합니다.

3. 업데이트된 **CR** 오브젝트를 다시 생성합니다.

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

검증

모니터링 중인 노드의 감사 로그 항목이 로깅 서버에서 수집한 로그 데이터 사이에 있는지 확인합니다.

/var/log/ovn/acl-audit-log.log에서 원래 감사 로그 항목을 찾아 로깅 서버의 해당 로그 항목과 비교합니다.

예를 들어 /var/log/ovn/acl-audit-log.log의 원본 로그 항목은 다음과 같을 수 있습니다.

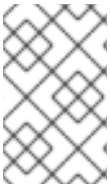
```
2021-07-06T08:26:58.687Z|00004|acl_log(ovn_pinctrl0)|INFO|name="verify-audit-
logging_deny-all", verdict=drop, severity=alert:
icmp,vlan_tci=0x0000,dl_src=0a:58:0a:81:02:12,dl_dst=0a:58:0a:81:02:14,nw_src=10
.129.2.18,nw_dst=10.129.2.20,nw_tos=0,nw_ecn=0,nw_ttl=64,icmp_type=8,icmp_code=0
```

로깅 서버에서 찾은 해당 OVN 감사 로그 항목은 다음과 같을 수 있습니다.

```
{
  "@timestamp" : "2021-07-06T08:26:58..687000+00:00",
  "hostname":"ip.abc.iternal",
  "level":"info",
  "message" : "2021-07-06T08:26:58.687Z|00004|acl_log(ovn_pinctrl0)|INFO|name=\"verify-
audit-logging_deny-all\", verdict=drop, severity=alert:
icmp,vlan_tci=0x0000,dl_src=0a:58:0a:81:02:12,dl_dst=0a:58:0a:81:02:14,nw_src=10.129.2.18,n
w_dst=10.129.2.20,nw_tos=0,nw_ecn=0,nw_ttl=64,icmp_type=8,icmp_code=0"
}
```

다음과 같습니다.

- @timestamp는 로그 항목의 타임스탬프입니다.
- hostname은 로그가 시작된 노드입니다.
- level은 로그 항목입니다.
- message는 원래 감사 로그 메시지입니다.



참고

Elasticsearch 서버에서 인덱스가 audit-00000으로 시작하는 로그 항목을 찾습니다.

문제 해결

1. OpenShift Container Platform 클러스터가 모든 사전 요구 사항을 충족하는지 확인합니다.

2. 절차를 완료했는지 확인합니다.

3. OVN 로그를 생성하는 노드가 활성화되어 있고 /var/log/ovn/acl-audit-log.log 파일이 있는지 확인합니다.

4. 문제가 있는지 Fluentd Pod 로그를 확인합니다.

추가 리소스

- 네트워크 정책 감사 로깅

7.16. 로그 전달 문제 해결

ClusterLogForwarder CR(사용자 정의 리소스)을 생성할 때 Red Hat OpenShift Logging Operator가 Fluentd Pod를 자동으로 재배포하지 않으면 Fluentd Pod를 삭제하여 강제로 재배포할 수 있습니다.

사전 요구 사항

- ClusterLogForwarder CR(사용자 정의 리소스) 오브젝트가 생성되어 있습니다.

절차

- Fluentd Pod를 삭제하여 강제로 재배포합니다.

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

8장. JSON 로깅 활성화

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

8.1. JSON 로그 구문 분석

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

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

구조화된 JSON 로그 항목 예

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

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

message 필드 예

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

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

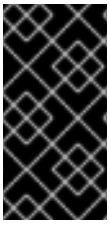
parse: json을 보여주는 코드 조각 예

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

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

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

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



중요

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

특정 로깅 플랫폼에 대한 **JSON** 로그를 구문 분석할 수 있도록 하려면 타사 시스템으로 로그 전달을 참조하십시오.

8.2. ELASTICSEARCH의 JSON 로그 데이터 구성

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

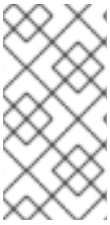
중요

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

구조 유형

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

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



참고

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

A structuredTypeKey: kubernetes.labels.<key> example

다음을 확인합니다.

-

클러스터에서 "apache" 및 "google"의 두 가지 형식으로 JSON 로그를 생성하는 애플리케이션 Pod를 실행하고 있습니다.

- 사용자는 logFormat=apache 및 logFormat=google를 사용하여 이러한 애플리케이션 pod에 레이블을 지정합니다.

- ClusterLogForwarder CR YAML 파일에서 다음 코드 조각을 사용합니다.

```
outputDefaults:
 elasticsearch:
   structuredTypeKey: kubernetes.labels.logFormat 1
   structuredTypeName: nologformat
pipelines:
- inputRefs: <application>
  outputRefs: default
  parse: json 2
```

1 Kubernetes logFormat 레이블로 구성된 키-값 쌍의 값을 사용합니다.

2 JSON 로그를 구문 분석할 수 있습니다.

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

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

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

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

A structuredTypeKey: openshift.labels.<key> example

ClusterLogForwarder CR YAML 파일에서 다음 코드 조각을 사용한다고 가정합니다.

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

1 OpenShift myLabel 레이블로 구성된 키-값 쌍의 값을 사용합니다.

2 myLabel 요소는 구조화된 로그 레코드에 문자열 값 myValue를 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 *인덱스*는 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 *형식*에만 사용합니다. 예를 들어 대부분의 Apache 애플리케이션은 LogApache와 같은 동일한 JSON 로그 형식과 구조화된 유형을 사용합니다.

8.3. ELASTICSEARCH 로그 저장소로 JSON 로그 전달

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

중요

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

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

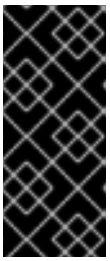
절차

1. 다음 조각을 ClusterLogForwarder CR YAML 파일에 추가합니다.

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

2. 선택 사항: structuredTypeKey 를 사용하여 이전 항목에 설명된 대로 로그 레코드 필드 중 하나를 지정하고 Elasticsearch에 대한 JSON 로그 데이터 구성을 지정합니다. 그렇지 않으면 다음 행을 제거합니다.

3. 선택 사항: structuredTypeName 을 사용하여 이전 항목에 설명된 <name> 을 지정하고 Elasticsearch에 대한 JSON 로그 데이터를 구성합니다. 그렇지 않으면 다음 행을 제거합니다.



중요

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

4. **inputRefs**의 경우 **application, infrastructure,** 또는 **audit** 등 해당 파이프라인을 사용하여 전달해야 하는 로그 유형을 지정합니다.

5. **parse: json** 요소를 파이프라인에 추가합니다.

6. **CR** 오브젝트를 생성합니다.

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

Red Hat OpenShift Logging Operator는 **Fluentd Pod**를 재배포합니다. **Pod**가 재배포되지 않으면 **Fluentd Pod**를 삭제하여 강제로 재배포할 수 있습니다.

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

추가 리소스

- 타사 시스템에 로그 전달

9장. 쿠버네티스 이벤트 수집 및 저장

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

이벤트 라우터는 모든 프로젝트에서 이벤트를 수집하여 STDOUT에 씁니다. Fluentd는 이러한 이벤트를 수집하여 OpenShift Container Platform Elasticsearch 인스턴스로 전달합니다. Elasticsearch는 이벤트를 인프라 인덱스에 인덱싱합니다.

중요

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

9.1. 이벤트 라우터 배포 및 구성

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

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

사전 요구 사항

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

프로세스

1. 이벤트 라우터용 템플릿을 생성합니다.

```
kind: Template
apiVersion: template.openshift.io/v1
metadata:
  name: eventrouter-template
```

```
  annotations:
    description: "A pod forwarding kubernetes events to OpenShift Logging stack."
    tags: "events,EFK,logging,cluster-logging"
objects:
  - kind: ServiceAccount 1
    apiVersion: v1
    metadata:
      name: eventrouter
      namespace: ${NAMESPACE}
  - kind: ClusterRole 2
    apiVersion: rbac.authorization.k8s.io/v1
    metadata:
      name: event-reader
    rules:
    - apiGroups: [""]
      resources: ["events"]
      verbs: ["get", "watch", "list"]
  - kind: ClusterRoleBinding 3
    apiVersion: rbac.authorization.k8s.io/v1
    metadata:
      name: event-reader-binding
    subjects:
    - kind: ServiceAccount
      name: eventrouter
      namespace: ${NAMESPACE}
    roleRef:
      kind: ClusterRole
      name: event-reader
  - kind: ConfigMap 4
    apiVersion: v1
    metadata:
      name: eventrouter
      namespace: ${NAMESPACE}
    data:
      config.json: |-
        {
          "sink": "stdout"
        }
  - kind: Deployment 5
    apiVersion: apps/v1
    metadata:
      name: eventrouter
      namespace: ${NAMESPACE}
      labels:
        component: "eventrouter"
        logging-infra: "eventrouter"
        provider: "openshift"
    spec:
      selector:
        matchLabels:
          component: "eventrouter"
          logging-infra: "eventrouter"
          provider: "openshift"
      replicas: 1
      template:
```

```
      metadata:
        labels:
          component: "eventrouter"
          logging-infra: "eventrouter"
          provider: "openshift"
        name: eventrouter
      spec:
        serviceAccount: eventrouter
        containers:
          - name: kube-eventrouter
            image: ${IMAGE}
            imagePullPolicy: IfNotPresent
            resources:
              requests:
                cpu: ${CPU}
                memory: ${MEMORY}
            volumeMounts:
            - name: config-volume
              mountPath: /etc/eventrouter
        volumes:
          - name: config-volume
            configMap:
              name: eventrouter
parameters:
  - name: IMAGE 6
    displayName: Image
    value: "registry.redhat.io/openshift-logging/eventrouter-rhel8:v0.3"
  - name: CPU 7
    displayName: CPU
    value: "100m"
  - name: MEMORY 8
    displayName: Memory
    value: "128Mi"
  - name: NAMESPACE
    displayName: Namespace
    value: "openshift-logging" 9
```

1

openshift-logging 프로젝트에서 이벤트 라우터용 서비스 계정을 생성합니다.

2

클러스터의 이벤트를 모니터링할 **ClusterRole**을 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 **Pod**를 생성하고 구성합니다.

6

v0.3 과 같은 태그로 식별되는 이미지를 지정합니다.

7

이벤트 라우터 **Pod**에 할당할 최소 메모리 양을 지정합니다. 기본값은 **128Mi**입니다.

8

이벤트 라우터 **Pod**에 할당할 최소 **CPU** 양을 지정합니다. 기본값은 **100m**입니다.

9

오브젝트를 설치할 **openshift-logging** 프로젝트를 지정합니다.

2. 다음 명령을 사용하여 템플릿을 처리하고 적용합니다.

```
$ oc process -f <templatefile> | oc apply -n openshift-logging -f -
```

예를 들면 다음과 같습니다.

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

출력 예

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

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

a. **새 이벤트 라우터 Pod 보기:**

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

출력 예

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

b. **이벤트 라우터에서 수집한 이벤트 보기:**

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

예를 들면 다음과 같습니다.

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

출력 예

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

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

10장. OPENSHIFT LOGGING 업데이트

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

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

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

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

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

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

OpenShift Container Platform 4.7에서는 다음과 같이 이름이 변경되었습니다.

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

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

그런 다음 다음 operator를 업데이트합니다.

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

중요

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

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

사전 요구 사항

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

절차

1. OpenShift Elasticsearch Operator를 업데이트합니다.
   a. 웹 콘솔에서 Operator → 설치된 Operator를 클릭합니다.

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

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

d. 서브스크립션 → 채널을 클릭합니다.

e. 서브스크립션 업데이트 채널 변경 창에서 **5.0** 또는 **stable-5.x**을 선택하고 저장을 클릭합니다.

f. 몇 초 정도 기다린 후 **Operator** → 설치된 **Operator**를 클릭합니다.

OpenShift Elasticsearch Operator 버전이 **5.x.x**인지 확인합니다.

상태 필드가 성공으로 표시될 때까지 기다립니다.

2. **Cluster Logging Operator** 업데이트:

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

b. **openshift-logging** 프로젝트를 선택합니다.

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

d. 서브스크립션 → 채널을 클릭합니다.

e. 서브스크립션 업데이트 채널 변경 창에서 **5.0** 또는 **stable-5.x**을 선택하고 저장을 클릭합니다.

f. 몇 초 정도 기다린 후 **Operator** → 설치된 **Operator**를 클릭합니다.

Red Hat OpenShift Logging Operator 버전이 **5.0.x** 또는 **5.x.x**인지 확인합니다.

상태 필드가 성공으로 표시될 때까지 기다립니다.

3. 로깅 구성 요소를 확인합니다.

a. 모든 **Elasticsearch Pod**가 **ready** 상태인지 확인합니다.

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

출력 예

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

b. **Elasticsearch** 클러스터가 정상인지 확인합니다.

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

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

c. **Elasticsearch Cron** 작업이 생성되었는지 확인합니다.

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

```
$ oc get cronjob
```

```
NAME                     SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
elasticsearch-im-app     */15 * * * *   False     0        <none>          56s
elasticsearch-im-audit   */15 * * * *   False     0        <none>          56s
elasticsearch-im-infra   */15 * * * *   False     0        <none>          56s
```

d. 로그 저장소가 **5.0** 또는 **5.x**로 업데이트되고 인덱스가 **green** 인지 확인합니다.

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

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

예 10.1. 인덱스가 녹색 상태인 샘플 출력

```
Tue Jun 30 14:30:54 UTC 2020
health status index                                 uuid                   pri rep
docs.count docs.deleted store.size pri.store.size
green  open   infra-000008
bnBvUFEXTWi92z3zWAzieQ   3 1      222195            0      289          144
green  open   infra-000004
rtDSzoqsSl6saisSK7Au1Q   3 1      226717            0      297          148
green  open   infra-000012
RSf_kUwDSR2xEuKRZMPqZQ   3 1      227623            0      295          147
green  open   .kibana_7
1SJdCqlZTPWlIAaOUd78yg   1 1          4            0        0            0
green  open   infra-000010
iXwL3bnqTuGEABbUDa6OVw   3 1      248368            0      317          158
green  open   infra-000009
YN9EsULWSNaxWeeNvOs0RA   3 1      258799            0      337          168
green  open   infra-000014
YP0U6R7FQ_GVQVQZ6Yh9Ig   3 1      223788            0      292          146
green  open   infra-000015
JRBbAbEmSMqK5X40df9HbQ   3 1      224371            0      291          145
green  open   .orphaned.2020.06.30
n_xQC2dWQzConkvQqei3YA   3 1          9            0        0            0
green  open   infra-000007
llkkAVSzSOmosWTSAJM_hg   3 1      228584            0      296          148
green  open   infra-000005
d9BoGQdiQASsS3BBFm2iRA   3 1      227987            0      297          148
green  open   infra-000003                                 1-
goREK1QUKlQPAIVkWVaQ   3 1      226719            0      295          147
green  open   .security
zeT65uOuRTKZMjg_bbUc1g   1 1          5            0        0            0
green  open   .kibana-377444158_kubeadmin
wvMhDwJkR-mRZQO84K0gUQ   3 1          1            0        0            0
green  open   infra-000006                                5H-
KBSXGQKiO7hdapDE23g   3 1      226676            0      295          147
green  open   infra-000001                                eH53BQ-
bSxSWR5xYZB6lVg   3 1      341800            0      443          220
green  open   .kibana-6
RVp7TemSSemGJcsSUmuf3A   1 1          4            0        0            0
```

```
green open infra-000011
J7XWBauWSTe0jnzX02fU6A 3 1 226100 0 293 146
green open app-000001
axSAFfONQDmKwatkjPXdtw 3 1 103186 0 126 57
green open infra-000016
m9c1iRLtStWSF1GopaRyCg 3 1 13685 0 19 9
green open infra-000002 Hz6WvINtTvKcQzw-
ewmbYg 3 1 228994 0 296 148
green open infra-000013 KR9mMFUpQl-
jraYtanylGw 3 1 228166 0 298 148
green open audit-000001
eERqLdLmQOiQDFES1LBATQ 3 1 0 0 0 0
```

e. 로그 수집기가 **5.0** 또는 **5.x**로 업데이트되었는지 확인합니다.

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

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

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

f. 로그 시각화 프로그램이 **Kibana CRD**를 사용하여 **5.0** 또는 **5.x**로 업데이트되었는지 확인합니다.

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

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

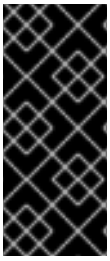
예 10.2. Kibana Pod가 준비된 샘플 출력

```
[
{
"clusterCondition": {
"kibana-5fdd766ffd-nb2jj": [
{
"lastTransitionTime": "2020-06-30T14:11:07Z",
"reason": "ContainerCreating",
"status": "True",
"type": ""
},
{
"lastTransitionTime": "2020-06-30T14:11:07Z",
"reason": "ContainerCreating",
"status": "True",
"type": ""
```

```
}
]
},
"deployment": "kibana",
"pods": {
"failed": [],
"notReady": []
"ready": []
},
"replicaSets": [
"kibana-5fdd766ffd"
],
"replicas": 1
}
]
```


10.2. OPENSHIFT LOGGING을 현재 버전으로 업데이트

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



중요

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

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

사전 요구 사항

- OpenShift Container Platform 4.7 이상 버전이어야 합니다.
- OpenShift Logging 상태가 정상이여야 합니다.
  - 모든 Pod가 ready 상태입니다.

- **Elasticsearch** 클러스터는 정상입니다.

- **Elasticsearch** 및 **Kibana** 데이터가 백업됩니다.

절차

1. **OpenShift Elasticsearch Operator**를 업데이트합니다.

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

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

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

   d. 서브스크립션 → 채널을 클릭합니다.

   e. 서브스크립션 업데이트 채널 변경 창에서 **stable-5.x**을 선택하고 저장을 클릭합니다.

   f. 몇 초 정도 기다린 후 **Operator** → 설치된 **Operator**를 클릭합니다.

      OpenShift Elasticsearch Operator 버전이 **5.x.x**인지 확인합니다.

      상태 필드가 성공으로 표시될 때까지 기다립니다.

2. **Red Hat OpenShift Logging Operator**를 업데이트합니다.

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

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

c. **Red Hat OpenShift Logging Operator**를 클릭합니다.

d. 서브스크립션 → 채널을 클릭합니다.

e. 서브스크립션 업데이트 채널 변경 창에서 **stable-5.x**을 선택하고 저장을 클릭합니다.

f. 몇 초 정도 기다린 후 **Operator** → 설치된 **Operator**를 클릭합니다.

Red Hat OpenShif Logging Operator 버전이 **5.x.x**인지 확인합니다.

상태 필드가 성공으로 표시될 때까지 기다립니다.

3. 로깅 구성 요소를 확인합니다.

a. 모든 **Elasticsearch Pod**가 **ready** 상태인지 확인합니다.

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

출력 예

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

b. **Elasticsearch** 클러스터가 정상인지 확인합니다.

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

```
{
  "cluster_name" : "elasticsearch",
```

```
  "status" : "green",
}
```

c.

Elasticsearch Cron 작업이 생성되었는지 확인합니다.

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

```
$ oc get cronjob
```

```
NAME                     SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
elasticsearch-im-app     */15 * * * *   False     0        <none>          56s
elasticsearch-im-audit   */15 * * * *   False     0        <none>          56s
elasticsearch-im-infra   */15 * * * *   False     0        <none>          56s
```

d.

로그 저장소가 5.x로 업데이트되고 인덱스가 green인지 확인합니다.

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

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

예 10.3. 인덱스가 녹색 상태인 샘플 출력

```
Tue Jun 30 14:30:54 UTC 2020
health status index                                    uuid                   pri rep
docs.count docs.deleted store.size pri.store.size
green  open   infra-000008
bnBvUFEXTWi92z3zWAzieQ   3 1      222195          0      289          144
green  open   infra-000004
rtDSzoqsSl6saisSK7Au1Q   3 1      226717          0      297          148
green  open   infra-000012
RSf_kUwDSR2xEuKRZMPqZQ   3 1      227623          0      295          147
green  open   .kibana_7
1SJdCqlZTPWlIAaOUd78yg   1 1          4          0        0            0
green  open   infra-000010
iXwL3bnqTuGEABbUDa6OVw   3 1      248368          0      317          158
green  open   infra-000009
YN9EsULWSNaxWeeNvOs0RA   3 1      258799          0      337          168
green  open   infra-000014
YP0U6R7FQ_GVQVQZ6Yh9lg   3 1      223788          0      292          146
green  open   infra-000015
JRBbAbEmSMqK5X40df9HbQ   3 1      224371          0      291          145
green  open   .orphaned.2020.06.30
n_xQC2dWQzConkvQqei3YA   3 1          9          0        0            0
green  open   infra-000007
llkkAVSzSOmosWTSAJM_hg   3 1      228584          0      296          148
green  open   infra-000005
d9BoGQdiQASsS3BBFm2iRA   3 1      227987          0      297          148
```

```
green  open   infra-000003                                 1-
goREK1QUKlQPAIVkWVaQ   3 1      226719          0      295          147
green  open   .security
zeT65uOuRTKZMjg_bbUc1g   1 1          5          0        0            0
green  open   .kibana-377444158_kubeadmin
wvMhDwJkR-mRZQO84K0gUQ   3 1          1          0        0            0
green  open   infra-000006                                 5H-
KBSXGQKiO7hdapDE23g   3 1      226676          0      295          147
green  open   infra-000001                                 eH53BQ-
bSxSWR5xYZB6lVg   3 1      341800          0      443          220
green  open   .kibana-6
RVp7TemSSemGJcsSUmuf3A   1 1          4          0        0            0
green  open   infra-000011
J7XWBauWSTe0jnzX02fU6A   3 1      226100          0      293          146
green  open   app-000001
axSAFfONQDmKwatkjPXdtw   3 1      103186          0      126           57
green  open   infra-000016
m9c1iRLtStWSF1GopaRyCg   3 1       13685          0       19            9
green  open   infra-000002                                 Hz6WvINtTvKcQzw-
ewmbYg   3 1      228994          0      296          148
green  open   infra-000013                                 KR9mMFUpQl-
jraYtanylGw   3 1      228166          0      298          148
green  open   audit-000001
eERqLdLmQOiQDFES1LBATQ   3 1          0          0        0            0
```

e. 로그 수집기가 **5.x**로 업데이트되었는지 확인합니다.

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

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

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

f. 로그 시각화 프로그램이 **Kibana CRD**를 사용하여 **5.x**로 업데이트되었는지 확인합니다.

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

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

예 10.4. Kibana Pod가 준비된 샘플 출력

```
[
{
"clusterCondition": {
"kibana-5fdd766ffd-nb2jj": [
```

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

11장. 클러스터 대시보드 보기

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

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

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

참고

더 자세한 데이터를 보려면 대시보드에서 Grafana UI 링크를 클릭하여 Grafana 대시보드를 시작합니다. Grafana는 OpenShift 클러스터 모니터링과 함께 제공됩니다.

11.1. ELASTISEARCH 및 OPENSHIFT LOGGING 대시보드에 액세스

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

프로세스

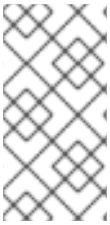
대시보드를 시작하려면 다음을 수행합니다.

1. OpenShift Container Platform 웹 콘솔에서 모니터링 → 대시보드를 클릭합니다.
2. 대시보드 페이지의 대시보드 메뉴에서 로깅/Elasticsearch 노드 또는 OpenShift 로깅 을 선택합니다.

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

   여러 데이터 차트를 보여주는 적절한 대시보드가 표시됩니다.

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



참고

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

대시보드 차트에 대한 정보는 **OpenShift** 로깅 대시보드 정보 및 로깅/**Elastisearch** 노드 대시보드 정보를 참조하십시오.

11.2. 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 11.1. OpenShift 로깅 차트

지표	설명
Elastic 클러스터 상태	현재 Elasticsearch 상태: ● 온라인 - Elasticsearch 인스턴스가 온라인 상태임을 나타냅니다. ● 오프라인 - Elasticsearch 인스턴스가 오프라인 상태임을 나타냅니다.
Elastic 노드	Elasticsearch 인스턴스의 총 Elasticsearch 노드 수입니다.
Elastic Shard	Elasticsearch 인스턴스의 총 Elasticsearch shard 수입니다.
Elastic 문서	Elasticsearch 인스턴스의 총 Elasticsearch 문서 수입니다.
디스크의 총 인덱스 크기	Elasticsearch 인덱스에 사용 중인 총 디스크 공간입니다.
Elastic 보류 작업	인덱스 생성, 인덱스 매핑, shard 할당 또는 shard 오류와 같이 완료되지 않은 Elasticsearch 변경의 총 수입니다.

지표	설명
Elastic JVM GC 시간	JVM이 클러스터에서 Elasticsearch 가비지 수집 작업을 실행하는 데 소비한 시간입니다.
Elastic JVM GC 속도	JVM이 초당 가비지 활동을 실행한 총 횟수입니다.
Elastic 쿼리/가져오기 대기 시간 합계	• 쿼리 대기 시간: 각 Elasticsearch 검색 쿼리를 실행하는 데 걸리는 평균 시간입니다. • 가져오기 대기 시간: 각 Elasticsearch 검색 쿼리에서 데이터를 가져오는 데 걸리는 평균 시간입니다. 가져오기 대기 시간은 일반적으로 쿼리 대기 시간보다 더 짧습니다. 가져오기 대기 시간이 지속적으로 증가하는 경우 느린 디스크, 데이터 보강 또는 결과가 너무 많은 대규모 요청을 나타낼 수 있습니다.
Elastic 쿼리 속도	각 Elasticsearch 노드에 대해 Elasticsearch 인스턴스에 대해 실행된 초당 총 쿼리입니다.
CPU	Elasticsearch, Fluentd 및 Kibana에서 사용하는 CPU 양(각 구성 요소에 대해 표시됨).
사용된 Elastic JVM 힙	사용된 JVM 메모리 양입니다. 정상 클러스터에서 그래프는 JVM 가비지 수집에 의해 메모리가 해제됨에 따라 정기적으로 감소를 표시합니다.
Elasticsearch 디스크 사용량	각 Elasticsearch 노드에 대해 Elasticsearch 인스턴스에서 사용하는 총 디스크 공간입니다.
사용 중인 파일 설명자	Elasticsearch, Fluentd 및 Kibana에서 사용하는 총 파일 설명자 수입니다.
FluentD 방출 수	Fluentd 기본 출력에 대한 초당 총 Fluentd 메시지 수 및 기본 출력에 대한 재시도 횟수입니다.
FluentD 버퍼 가용성	청크에 사용할 수 있는 Fluentd 버퍼의 백분율입니다. 가득 찬 버퍼는 Fluentd가 수신된 로그 수를 처리할 수 없음을 나타낼 수 있습니다.
Elastic rx 바이트	Elasticsearch가 FluentD, Elasticsearch 노드 및 기타 소스에서 수신한 총 바이트 수입니다.
Elastic 인덱스 실패율	Elasticsearch 인덱스가 실패하는 초당 총 횟수입니다. 높은 비율은 인덱싱 문제를 나타낼 수 있습니다.
FluentD 출력 오류율	FluentD가 로그를 출력할 수 없는 초당 총 횟수입니다.

11.3. 로깅/ELASTICSEARCH 노드 대시보드의 차트

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

Elasticsearch 상태

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

표 11.2. Elasticsearch 상태 필드

지표	설명
클러스터 상태	Elasticsearch 녹색, 노란색 및 빨간색 상태를 사용하여 선택한 기간 동안의 클러스터 상태: ● 0 - Elasticsearch 인스턴스가 녹색 상태임을 나타냅니다. 이는 모든 shard가 할당되었음을 의미합니다. ● 1 - Elasticsearch 인스턴스가 노란색 상태임을 나타냅니다. 이는 하나 이상의 shard에 대한 복제본 shard가 할당되지 않았음을 의미합니다. ● 2 - Elasticsearch 인스턴스가 빨간색 상태임을 나타냅니다. 이는 하나 이상의 기본 shard와 해당 복제본이 할당되지 않았음을 의미합니다.
클러스터 노드	클러스터의 총 Elasticsearch 노드 수입니다.
클러스터 데이터 노드	클러스터에 있는 Elasticsearch 데이터 노드의 수입니다.
클러스터 보류 작업	완료되지 않고 클러스터 큐에서 대기 중인 클러스터 상태 변경 수(예: 인덱스 생성, 인덱스 삭제 또는 shard 할당)입니다. 증가 추세는 클러스터가 변경 사항을 따라갈 수 없음을 나타냅니다.

Elasticsearch 클러스터 인덱스 shard 상태

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

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

인덱스 **shard**는 데이터의 물리적 표현인 인덱스 세그먼트라고 하는 여러 개의 작은 내부 블록으로 구성됩니다. 인덱스 세그먼트는 **Lucene**이 새로 인덱싱된 데이터를 커밋할 때 생성되는 비교적 작고 변경 불가능한 **Lucene** 인덱스입니다. **Elasticsearch**에서 사용하는 검색 라이브러리인 **Lucene**은 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 **Lucene**은 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 **shard**에 대한 다음 차트가 포함되어 있습니다.

표 11.3. Elasticsearch 클러스터 shard 상태 차트

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

Elasticsearch 노드 지표

각 **Elasticsearch** 노드에는 작업을 처리하는 데 사용할 수 있는 한정된 양의 리소스가 있습니다. 모든 리소스가 사용되고 **Elasticsearch**가 새 작업을 수행하려고 하면 **Elasticsearch**는 일부 리소스를

사용할 수 있을 때까지 작업을 큐에 넣습니다.

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

표 11.4. Elasticsearch 노드 지표 차트

지표	설명
ThreadPool 작업	작업 유형별로 표시되는 개별 큐의 대기 작업 수입니다. 큐에 작업이 장기간 누적되면 노드 리소스 부족 또는 기타 문제가 있을 수 있습니다.
CPU 사용량	선택한 Elasticsearch 노드에서 사용 중인 CPU 양(호스트 컨테이너에 할당된 총 CPU의 백분율)입니다.
메모리 사용량	선택한 Elasticsearch 노드에서 사용 중인 메모리 양입니다.
디스크 사용량	선택한 Elasticsearch 노드에서 인덱스 데이터 및 메타데이터에 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.
검색 대기 시간	선택한 Elasticsearch 노드에서 검색 요청을 완료하는 데 걸린 시간입니다. 검색 대기 시간은 여러 요인의 영향을 받을 수 있습니다. 대기 시간 증가는 인스턴스의 리소스 용량이 부족함을 나타냅니다.
문서 수(복제본 포함)	노드에 할당된 기본 shard와 복제본 shard 모두에 저장된 문서를 포함하여 선택한 Elasticsearch 노드에 저장된 Elasticsearch 문서 수입니다.
문서 삭제 비율	선택한 Elasticsearch 노드에 할당된 인덱스 shard에서 삭제되는 Elasticsearch 문서의 수입니다.
문서 병합 비율	선택한 Elasticsearch 노드에 할당된 인덱스 shard에서 병합되는 Elasticsearch 문서의 수입니다.

Elasticsearch 노드 필드 데이터

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

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

표 11.5. Elasticsearch 노드 필드 데이터 차트

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

Elasticsearch 노드 쿼리 캐시

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

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

표 11.6. Elasticsearch 노드 쿼리 차트

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

Elasticsearch 인덱스 제한

문서를 인덱싱할 때 Elasticsearch는 데이터의 물리적 표현인 인덱스 세그먼트에 문서를 저장합니다. 동시에 Elasticsearch는 리소스 사용을 최적화하기 위해 주기적으로 작은 세그먼트를 큰 세그먼

트로 병합합니다. 인덱싱이 세그먼트 병합 기능보다 빠르면 병합 프로세스가 충분히 빨리 완료되지 않아 검색 및 성능에 문제가 발생할 수 있습니다. 이러한 상황을 방지하기 위해 Elasticsearch는 일반적으로 인덱싱에 할당된 스레드 수를 단일 스레드로 줄여 인덱싱을 제한합니다.

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

표 11.7. 인덱스 제한 차트

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

노드 JVM 힙 통계

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

표 11.8. JVM 힙 통계 차트

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

12장. 로깅 문제 해결

12.1. OPENSHIFT LOGGING 상태 보기

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

12.1.1. Red Hat OpenShift Logging Operator의 상태 보기

Red Hat OpenShift Logging Operator의 상태를 볼 수 있습니다.

사전 요구 사항

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

1. openshift-logging 프로젝트로 변경합니다.

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

2. OpenShift Logging 상태를 보려면 다음을 수행합니다.

   a. OpenShift Logging 상태를 가져옵니다.

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

출력 예

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

....

status: 1
 collection:
   logs:
     fluentdStatus:
```

```
      daemonSet: fluentd 2
      nodes:
        fluentd-2rhqp: ip-10-0-169-13.ec2.internal
        fluentd-6fgjh: ip-10-0-165-244.ec2.internal
        fluentd-6l2ff: ip-10-0-128-218.ec2.internal
        fluentd-54nx5: ip-10-0-139-30.ec2.internal
        fluentd-flpnn: ip-10-0-147-228.ec2.internal
        fluentd-n2frh: ip-10-0-157-45.ec2.internal
      pods:
        failed: []
        notReady: []
        ready:
        - fluentd-2rhqp
        - fluentd-54nx5
        - fluentd-6fgjh
        - fluentd-6l2ff
        - fluentd-flpnn
        - fluentd-n2frh
  logstore: 3
    elasticsearchStatus:
    - ShardAllocationEnabled:  all
      cluster:
        activePrimaryShards:   5
        activeShards:          5
        initializingShards:    0
        numDataNodes:          1
        numNodes:              1
        pendingTasks:          0
        relocatingShards:      0
        status:                green
        unassignedShards:      0
      clusterName:             elasticsearch
      nodeConditions:
        elasticsearch-cdm-mkkdys93-1:
      nodeCount:  1
      pods:
        client:
          failed:
          notReady:
          ready:
          - elasticsearch-cdm-mkkdys93-1-7f7c6-mjm7c
        data:
          failed:
          notReady:
          ready:
          - elasticsearch-cdm-mkkdys93-1-7f7c6-mjm7c
        master:
          failed:
          notReady:
          ready:
          - elasticsearch-cdm-mkkdys93-1-7f7c6-mjm7c
  visualization:  4
    kibanaStatus:
    - deployment: kibana
      pods:
        failed: []
```

```
  notReady: []
  ready:
  - kibana-7fb4fd4cc9-f2nls
replicaSets:
- kibana-7fb4fd4cc9
replicas: 1
```

1 출력에서 클러스터 상태 필드가 상태 스탠자에 나타납니다.

2 Fluentd Pod에 대한 정보.

3 Elasticsearch 클러스터 건강, 녹색, 노란색 또는 빨간색을 포함한 Elasticsearch Pod에 대한 정보입니다.

4 Kibana Pod에 대한 정보.

12.1.1.1. 상태 메시지 예

다음은 OpenShift Logging 인스턴스의 Status.Nodes 섹션에 있는 일부 상태 메시지의 예입니다.

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

출력 예

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

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

다음과 유사한 상태 메시지는 노드가 구성된 높은 워터마크를 초과했으며 **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 **CR**의 **Elasticsearch** 노드 선택기가 클러스터의 노드와 일치하지 않음을 나타냅니다.

출력 예

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

```
    Node Conditions:
      elasticsearch-cdm-mkkdys93-1:
        Last Transition Time:  2019-06-26T03:37:32Z
        Message:               0/5 nodes are available: 5 node(s) didn't match node selector.
        Reason:                Unschedulable
        Status:                True
        Type:                  Unschedulable
      elasticsearch-cdm-mkkdys93-2:
    Node Count:  2
    Pods:
      Client:
        Failed:
        Not Ready:
          elasticsearch-cdm-mkkdys93-1-75dd69dccd-f7f49
          elasticsearch-cdm-mkkdys93-2-67c64f5f4c-n58vl
        Ready:
      Data:
        Failed:
        Not Ready:
          elasticsearch-cdm-mkkdys93-1-75dd69dccd-f7f49
          elasticsearch-cdm-mkkdys93-2-67c64f5f4c-n58vl
        Ready:
      Master:
        Failed:
        Not Ready:
          elasticsearch-cdm-mkkdys93-1-75dd69dccd-f7f49
          elasticsearch-cdm-mkkdys93-2-67c64f5f4c-n58vl
        Ready:
```

다음과 유사한 상태 메시지는 요청한 **PVC**가 **PV**에 바인딩할 수 없음을 나타냅니다.

출력 예

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

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

출력 예

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

12.1.2. OpenShift Logging 구성 요소의 상태 보기

여러 **OpenShift Logging** 구성 요소의 상태를 볼 수 있습니다.

사전 요구 사항

- **OpenShift Logging** 및 **Elasticsearch**가 설치되어 있어야 합니다.

프로세스

1. **openshift-logging** 프로젝트로 변경합니다.

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

2. **OpenShift Logging** 환경의 상태 보기:

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

출력 예

-

```
Name:                 cluster-logging-operator

....

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

....

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

3. **OpenShift Logging 복제본 세트 상태 보기:**

   a. 복제본 세트의 이름을 가져옵니다.

   출력 예

   ```
   $ oc get replicaset
   ```

   출력 예

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

b. 복제본 세트의 상태를 가져옵니다.

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

출력 예

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

....

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

....

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

12.2. ELASTICSEARCH 로그 저장소의 상태 보기

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

12.2.1. 로그 저장소의 상태 보기

로그 저장소의 상태를 볼 수 있습니다.

사전 요구 사항

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

1. openshift-logging 프로젝트로 변경합니다.

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

2. 상태를 보려면 다음을 수행합니다.

a. 로그 저장소 인스턴스의 이름을 가져옵니다.

```
$ oc get Elasticsearch
```

출력 예

```
NAME            AGE
elasticsearch   5h9m
```

b. 로그 저장소 상태를 가져옵니다.

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

예를 들면 다음과 같습니다.

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

출력에는 다음과 유사한 정보가 포함됩니다.

출력 예

```
status: 1
 cluster: 2
   activePrimaryShards: 30
   activeShards: 60
   initializingShards: 0
   numDataNodes: 3
   numNodes: 3
   pendingTasks: 0
   relocatingShards: 0
```

```
  status: green
  unassignedShards: 0
clusterHealth: ""
conditions: [] (3)
nodes: (4)
- deploymentName: elasticsearch-cdm-zjf34ved-1
  upgradeStatus: {}
- deploymentName: elasticsearch-cdm-zjf34ved-2
  upgradeStatus: {}
- deploymentName: elasticsearch-cdm-zjf34ved-3
  upgradeStatus: {}
pods: (5)
  client:
    failed: []
    notReady: []
    ready:
    - elasticsearch-cdm-zjf34ved-1-6d7fbf844f-sn422
    - elasticsearch-cdm-zjf34ved-2-dfbd988bc-qkzjz
    - elasticsearch-cdm-zjf34ved-3-c8f566f7c-t7zkt
  data:
    failed: []
    notReady: []
    ready:
    - elasticsearch-cdm-zjf34ved-1-6d7fbf844f-sn422
    - elasticsearch-cdm-zjf34ved-2-dfbd988bc-qkzjz
    - elasticsearch-cdm-zjf34ved-3-c8f566f7c-t7zkt
  master:
    failed: []
    notReady: []
    ready:
    - elasticsearch-cdm-zjf34ved-1-6d7fbf844f-sn422
    - elasticsearch-cdm-zjf34ved-2-dfbd988bc-qkzjz
    - elasticsearch-cdm-zjf34ved-3-c8f566f7c-t7zkt
shardAllocationEnabled: all
```

(1)

출력에서 클러스터 상태 필드가 상태 스탠자에 나타납니다.

(2)

로그 저장소의 상태:

- 활성 기본 **shard** 수입니다.
- 활성 **shard** 수입니다.

- 초기화 중인 **shard** 수입니다.
- 로그 저장소 데이터 노드 수입니다.
- 총 로그 저장소 노드 수입니다.
- 보류 중인 작업 수입니다.
- 로그 저장소 상태는 녹색, 빨간색, 노란색입니다.
- 할당되지 않은 **shard** 수

3 존재하는 경우 모든 상태 조건. 로그 저장소 상태는 **Pod**를 배치할 수 없는 경우 스케줄러의 사유를 나타냅니다. 다음 조건과 관련된 모든 이벤트가 표시됩니다.

- 컨테이너 로그 저장소 및 프록시 컨테이너를 기다리는 중입니다.
- 컨테이너 로그 저장소 및 프록시 컨테이너 모두에 대해 종료되었습니다.
- **Pod** 예약 불가. 또한 여러 가지 문제에 대한 조건이 표시됩니다(조건 메시지 예 참조).

4 **upgradeStatus**가 있는 클러스터의 로그 저장소 노드.

5 **'failed`**, **notReady** 또는 **ready** 상태 아래에 나열된 클러스터의 로그를 저장 클라이언트, 데이터 및 마스터 **Pod**.

12.2.1.1. 상태 메시지 예

다음은 Elasticsearch 인스턴스의 상태 섹션에 있는 일부 조건 메시지의 예입니다.

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

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

다음 상태 메시지는 노드가 구성된 높은 워터마크를 초과했으며 shard가 다른 노드로 재배치됨을 나타냅니다.

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

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

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

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

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

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

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

이 상태 메시지는 클러스터에 컨트롤 플레인 노드 (마스터 노드라고도 함)가 너무 많음을 나타냅니다.

```
status:
 clusterHealth: green
 conditions:
   - lastTransitionTime: '2019-04-17T20:12:34Z'
     message: >-
       Invalid master nodes count. Please ensure there are no more than 3 total
       nodes with master roles
     reason: Invalid Settings
     status: 'True'
     type: InvalidMasters
```

다음 상태 메시지는 **Elasticsearch** 스토리지가 변경 작업을 지원하지 않음을 나타냅니다.

예를 들면 다음과 같습니다.

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

```
reason: StorageStructureChangeIgnored
status: 'True'
type: StorageStructureChangeIgnored
```

reason 및 type 필드는 지원되지 않는 변경 유형을 지정합니다.

StorageClassNameChangeIgnored

스토리지 클래스 이름에 대한 지원되지 않는 변경 사항입니다.

StorageSizeChangeIgnored

스토리지 크기에 대한 지원되지 않는 변경 사항입니다.

StorageStructureChangeIgnored

임시 스토리지 구조와 영구저장장치 구조 간에는 지원되지 않는 변경 사항입니다.

중요

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

12.2.2. 로그 저장소 구성 요소의 상태 보기

여러 로그 저장소 구성 요소의 상태를 볼 수 있습니다.

Elasticsearch 인덱스

Elasticsearch 인덱스의 상태를 볼 수 있습니다.

1. **Elasticsearch Pod**의 이름을 가져옵니다.

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

출력 예

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

2. 인덱스의 상태를 가져옵니다.

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

출력 예

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

green  open   infra-000002
S4QANnf1QP6NgCegfnrnbQ   3   1    119926          0      157          78
green  open   audit-000001
8_EQx77iQCSTzFOXtxRqFw   3   1        0          0        0           0
green  open   .security                                  iDjscH7aSUGhIdq0LheLBQ
1   1         5          0        0           0
green  open   .kibana_-377444158_kubeadmin
yBywZ9GfSrKebz5gWBZbjw   3   1        1          0        0           0
green  open   infra-000001
z6Dpe__ORgiopEpW6Yl44A   3   1    871000          0      874         436
green  open   app-000001                                  hIrazQCeSISewG3c2VIvsQ
3   1      2453          0        3           1
green  open   .kibana_1                                  JCitcBMSQxKOvIq6iQW6wg
1   1         0          0        0           0
green  open   .kibana_-1595131456_user1                        gIYFIEGRRe-
ka0W3okS-mQ   3   1        1          0        0           0
```

로그 저장소 **Pod**

로그 저장소를 호스팅하는 **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-
```

```
rhel8@sha256:4265742c7cdd85359140e2d7d703e4311b6497eec7676957f455d6908e7b1c25
    Readiness:  exec [/usr/share/elasticsearch/probe/readiness.sh] delay=10s timeout=30s period=5s #success=1 #failure=3

....

Events:         <none>
```

12.2.3. Elasticsearch 클러스터 상태

OpenShift Container Platform 웹 콘솔의 모니터링 섹션에 있는 Grafana 대시보드에는 Elasticsearch 클러스터의 상태가 표시됩니다.

OpenShift Elasticsearch 클러스터의 상태를 보려면 OpenShift Container Platform 웹 콘솔의 모니터링 섹션의 Grafana 대시보드를 <cluster_url>/monitoring/dashboards/grafana-dashboard-cluster-logging 에서 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",
```

12.3. OPENSHIFT LOGGING 경고 이해

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

12.3.1. 로깅 수집기 경고 보기

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

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

프로세스

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

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

추가 리소스

- 경고 UI에 대한 자세한 내용은 경고 관리를 참조하십시오.

12.3.2. 로깅 수집기 경고 정보

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

표 12.1. Fluentd Prometheus 경고

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

12.3.3. Elasticsearch 경고 규칙 정보

이러한 경고 규칙을 Prometheus에서 볼 수 있습니다.

표 12.2. 경고 규칙

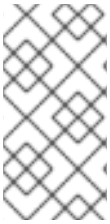
경고	설명	심각도
ElasticsearchClusterNotHealthy	클러스터 상태가 2분 이상 빨간색이었습니다. 클러스터가 쓰기를 허용하지 않거나 shard가 누락되었거나 마스터 노드가 아직 선택되지 않았을 수 있습니다.	심각
ElasticsearchClusterNotHealthy	클러스터 상태가 최소 20분 동안 노란색이었습니다. 일부 shard 복제본이 할당되지 않았습니다.	경고
ElasticsearchDiskSpaceRunningLow	클러스터는 향후 6시간 내에 디스크 공간이 부족할 것으로 예상됩니다.	심각
ElasticsearchHighFileDescriptorUsage	클러스터는 다음 시간 내에 파일 설명자가 없을 것으로 예상됩니다.	경고
ElasticsearchJVMHeapUseHigh	지정된 노드의 JVM 힙 사용량이 높습니다.	경고
ElasticsearchNodeDiskWatermarkReached	디스크 여유 공간이 부족하여 지정된 노드가 낮은 워터마크에 도달했습니다. 더 이상 shard를 이 노드에 할당할 수 없습니다. 노드에 디스크 공간을 추가하는 것을 고려해야 합니다.	정보
ElasticsearchNodeDiskWatermarkReached	디스크 여유 공간이 부족하여 지정된 노드가 높은 워터마크에 도달했습니다. 일부 shard는 가능한 경우 다른 노드에 다시 할당됩니다. 노드에 디스크 공간을 더 추가하거나 이 노드에 할당된 오래된 인덱스를 삭제하십시오.	경고
ElasticsearchNodeDiskWatermarkReached	디스크 여유 공간이 부족하여 지정된 노드가 플러드 워터마크에 도달했습니다. 이 노드에 할당된 shard가 있는 모든 인덱스에는 읽기 전용 블록이 적용됩니다. 디스크 사용량이 높은 워터마크 아래로 떨어지면 인덱스 블록을 수동으로 해제해야 합니다.	심각
ElasticsearchJVMHeapUseHigh	지정된 노드의 JVM 힙 사용량이 너무 높습니다.	경고
ElasticsearchWriteRequestsRejectionJumps	Elasticsearch의 지정된 노드에서 쓰기 거부가 증가하고 있습니다. 이 노드는 인덱싱 속도를 따라가지 못할 수 있습니다.	경고
AggregatedLoggingSystemCPUHigh	지정된 노드의 시스템에서 사용하는 CPU가 너무 높습니다.	경고
ElasticsearchProcessCPUHigh	지정된 노드에서 Elasticsearch가 사용하는 CPU가 너무 높습니다.	경고

12.4. RED HAT 지원을 위한 로깅 데이터 수집

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

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

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



참고

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

12.4.1. must-gather 툴 정보

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

OpenShift Logging 환경의 경우 **must-gather**는 다음 정보를 수집합니다.

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

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

12.4.2. 사전 요구 사항

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

12.4.3. OpenShift Logging 데이터 수집

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

프로세스

must-gather로 OpenShift Logging 정보를 수집하려면 다음을 수행하십시오.

1. must-gather 정보를 저장하려는 디렉터리로 이동합니다.

2. OpenShift Logging 이미지에 대해 oc adm must-gather 명령을 실행합니다.

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

must-gather 툴에서 현재 디렉터리 내에 must-gather.local로 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 Portal에서 해당 지원 사례에 압축 파일을 첨부합니다.

12.5. 심각한 경고 문제 해결

12.5.1. Elasticsearch 클러스터 상태가 빨간색임

하나 이상의 기본 shard와 해당 복제본이 노드에 할당되지 않습니다.

문제 해결

1. Elasticsearch 클러스터 상태를 확인하고 클러스터 status가 빨간색인지 확인합니다.

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

2. 클러스터에 참여한 노드를 나열합니다.

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

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

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

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

   a. Elasticsearch에 선택한 컨트롤 플레인 노드가 있는지 확인합니다.

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

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

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

   c. 클러스터에 참여하지 않은 노드의 로그에서 문제가 있는지 검토합니다.

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

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

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

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

6. 보류 중인 작업이 있는지 확인합니다.

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

7. 보류 중인 작업이 있는 경우 상태를 모니터링합니다.

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

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

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

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

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

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

10. 어떤 인덱스가 아직 빨간색인지 확인합니다.

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

11. 인덱스가 빨간색이면 다음 단계를 수행하여 지웁니다.

a. 캐시를 지웁니다.

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

b. 최대 할당 재시도 횟수를 늘립니다.

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

c. 모든 스크롤 항목을 삭제합니다.

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

d. 시간 제한을 늘립니다.

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

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

a. 빨간색 인덱스 이름을 확인합니다.

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

b. 빨간색 인덱스를 삭제합니다.

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

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

a. Elasticsearch JVM 힙 사용량이 높은지 확인합니다.

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

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

b. CPU 사용률이 높은지 확인합니다.

추가 리소스

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

12.5.2. Elasticsearch 클러스터 상태가 노란색임

하나 이상의 기본 shard의 복제본 shard는 노드에 할당되지 않습니다.

문제 해결

1. ClusterLogging CR에서 nodeCount를 조정하여 노드 수를 늘립니다.

추가 리소스

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

12.5.3. Elasticsearch 노드 디스크 Low Watermark Reached

Elasticsearch는 낮은 워터마크에 도달하는 노드에 shard를 할당하지 않습니다.

문제 해결

1. Elasticsearch가 배포된 노드를 식별합니다.

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

2. unassigned shards가 있는지 확인합니다.

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

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

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

4. nodes.node_name.fs 필드를 확인하여 해당 노드에서 사용 가능한 디스크 공간을 확인합니다.

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

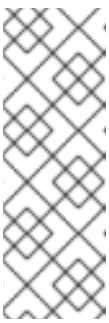
5. 모든 노드의 디스크 공간을 늘리십시오.

6. 디스크 공간을 늘릴 수 없는 경우 클러스터에 새 데이터 노드를 추가해 보십시오.

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

a. 현재 redundancyPolicy를 확인합니다.

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



참고

ClusterLogging CR을 사용하는 경우 다음을 입력합니다.

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

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

8. 이전 단계에서 문제가 해결되지 않으면 이전 인덱스를 삭제합니다.

a. **Elasticsearch**의 모든 인덱스의 상태를 확인합니다.

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

b. 삭제할 수 있는 이전 인덱스를 확인합니다.

c. 인덱스를 삭제합니다.

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

추가 리소스

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

12.5.4. Elasticsearch 노드 디스크 High Watermark Reached

Elasticsearch는 높은 워터마크에 도달한 노드에서 **shard**를 재배치하려고 합니다.

문제 해결

1. **Elasticsearch**가 배포된 노드를 식별합니다.

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

2. 각 노드의 디스크 공간을 확인합니다.

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

3. 클러스터가 재조정 중인지 확인합니다.

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

명령 출력에 **shard** 재배치가 표시되면 **High QCOWmark**가 초과된 것입니다. **High QCOWmark**의 기본값은 **90%**입니다.

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

4. 특정 노드에 **shard**를 할당하려면 일부 공간을 확보합니다.

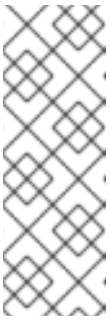
5. 모든 노드의 디스크 공간을 늘리십시오.

6. 디스크 공간을 늘릴 수 없는 경우 클러스터에 새 데이터 노드를 추가해 보십시오.

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

   a. 현재 **redundancyPolicy**를 확인합니다.

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

   참고

   ClusterLogging CR을 사용하는 경우 다음을 입력합니다.

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

   b. 클러스터 **redundancyPolicy**가 **SingleRedundancy** 보다 큰 경우

SingleRedundancy로 설정하고 이러한 변경 사항을 저장합니다.

8. 이전 단계에서 문제가 해결되지 않으면 이전 인덱스를 삭제합니다.

a. Elasticsearch의 모든 인덱스의 상태를 확인합니다.

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

b. 삭제할 수 있는 이전 인덱스를 확인합니다.

c. 인덱스를 삭제합니다.

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

추가 리소스

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

12.5.5. Elasticsearch 노드 디스크 Flood Watermark Reached

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

- 하나 이상의 shard가 노드에 할당됩니다.
- 하나 이상의 디스크가 플러드 단계를 초과합니다.

문제 해결

1. Elasticsearch 노드의 디스크 공간을 확인합니다.

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

nodes.node_name.fs 필드를 확인하여 해당 노드에서 사용 가능한 디스크 공간을 확인합니다.

2. 사용된 디스크 백분율이 **95%**를 초과하면 노드가 플러드 워터마크를 초과했음을 나타냅니다. 이 특정 노드에 할당된 **shard**에 대해 쓰기가 차단됩니다.

3. 모든 노드의 디스크 공간을 늘리십시오.

4. 디스크 공간을 늘릴 수 없는 경우 클러스터에 새 데이터 노드를 추가해 보십시오.

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

   a. 현재 **redundancyPolicy**를 확인합니다.

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

   참고

   ClusterLogging CR을 사용하는 경우 다음을 입력합니다.

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

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

6. 이전 단계에서 문제가 해결되지 않으면 이전 인덱스를 삭제합니다.

   a. **Elasticsearch**의 모든 인덱스의 상태를 확인합니다.

   -

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

b. 삭제할 수 있는 이전 인덱스를 확인합니다.

c. 인덱스를 삭제합니다.

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

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

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

추가 리소스

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

12.5.6. Elasticsearch JVM 힙 사용량이 높음

사용된 **Elasticsearch** 노드 **JVM** 힙 메모리는 **75%** 이상입니다.

문제 해결

힙 크기를 늘리는 것이 좋습니다.

12.5.7. 집계된 로깅 시스템 CPU가 높음

노드의 시스템 **CPU** 사용량이 높습니다.

문제 해결

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

12.5.8. Elasticsearch 프로세스 CPU가 높음

노드의 **Elasticsearch** 프로세스 **CPU** 사용량이 높습니다.

문제 해결

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

12.5.9. Elasticsearch 디스크 공간이 부족

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

문제 해결

1. **Elasticsearch** 노드의 디스크 공간을 가져옵니다.

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

2. 명령 출력에서 **nodes.node_name.fs** 필드를 확인하여 해당 노드의 사용 가능한 디스크 공간을 확인합니다.

3. 모든 노드의 디스크 공간을 늘리십시오.

4. 디스크 공간을 늘릴 수 없는 경우 클러스터에 새 데이터 노드를 추가해 보십시오.

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

   a. 현재 **redundancyPolicy**를 확인합니다.

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

참고

ClusterLogging CR을 사용하는 경우 다음을 입력합니다.

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

b. 클러스터 redundancyPolicy가 SingleRedundancy 보다 큰 경우 SingleRedundancy로 설정하고 이러한 변경 사항을 저장합니다.

6. 이전 단계에서 문제가 해결되지 않으면 이전 인덱스를 삭제합니다.

a. Elasticsearch의 모든 인덱스의 상태를 확인합니다.

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

b. 삭제할 수 있는 이전 인덱스를 확인합니다.

c. 인덱스를 삭제합니다.

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

추가 리소스

- 클러스터 로깅 사용자 정의 리소스 정보의 "샘플 ClusterLogging 사용자 정의 리소스 (CR)"에서 "redundancyPolicy"를 검색합니다.
- Elasticsearch 경고 규칙에서 "ElasticsearchDiskSpaceRunningLow"를 검색합니다.
- Elasticsearch 주제에서 "Free up or increase disk space"를 검색하고 빨간색 또는 노란색 클러스터 상태를 수정합니다.

12.5.10. Elasticsearch FileDescriptor 사용량이 높음

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

문제 해결

Elasticsearch 파일 설명자 항목에 설명된 대로 필요에 따라 각 노드의 max_file_descriptors 값을 확인하고 필요한 경우 구성합니다.

추가 리소스

- Elasticsearch 경고 규칙에서 "ElasticsearchHighFileDescriptorUsage"를 검색합니다.
- OpenShift Logging 대시보드에서 "File Descriptors In Use"를 검색합니다.

13장. OPENSHIFT LOGGING 설치 제거

OpenShift Container Platform 클러스터에서 OpenShift Logging을 제거할 수 있습니다.

13.1. OPENSHIFT CONTAINER PLATFORM에서 OPENSHIFT LOGGING 설치 삭제

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

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

사전 요구 사항

- OpenShift Logging 및 Elasticsearch가 설치되어 있어야 합니다.

프로세스

OpenShift Logging을 제거하려면 다음을 수행합니다.

1. OpenShift Container Platform 웹 콘솔을 사용하여 ClusterLogging CR을 제거합니다.
   a. 관리 → 사용자 정의 리소스 정의 페이지로 전환합니다.
   b. 사용자 정의 리소스 정의 페이지에서 ClusterLogging을 클릭합니다.
   c. 사용자 정의 리소스 정의 세부 정보 페이지에서 인스턴스를 클릭합니다.
   d. 인스턴스 옆에 있는 옵션 메뉴  를 클릭하고 ClusterLogging 삭제를 선택합니다.

2. 선택 사항: CRD(사용자 정의 리소스 정의)를 삭제합니다.

   a. 관리 → 사용자 정의 리소스 정의 페이지로 전환합니다.

   b. ClusterLogForwarder 옆에 있는 옵션 메뉴  를 클릭하고 사용자 정의 리소스 정의 삭제를 선택합니다.

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

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

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

   a. Operator → 설치된 Operator 페이지로 전환합니다.

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

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

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

a. 홈 → 프로젝트 페이지로 전환합니다.

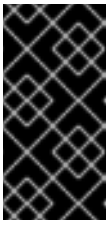
b. openshift-logging 프로젝트 옆에 있는 옵션 메뉴

를 클릭하고 프로젝트 삭제를 선택합니다.

c. 대화 상자에서 openshift-logging을 입력하여 삭제를 확인하고 삭제를 클릭합니다.

d. openshift-operators-redhat 프로젝트 옆에 있는 옵션 메뉴

를 클릭하고 프로젝트 삭제를 선택합니다.



중요

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

e. 대화 상자에서 openshift-operators-redhat을 입력하여 삭제를 확인하고 삭제를 클릭합니다.

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

6. 선택 사항: PVC를 유지하지 않으려면 삭제할 수 있습니다.

주의

PVC를 해제하거나 삭제하면 PV가 삭제되고 데이터 손실이 발생할 수 있습니다.

a. 스토리지 → 영구 볼륨 클레임 페이지로 전환합니다.

b. 각 PVC 옆에 있는 옵션 메뉴

를 클릭하고 영구 볼륨 클레임 삭제를 선택합니다.

c. 스토리지 공간을 복구하려면 PV를 삭제할 수 있습니다.

추가 리소스

- 수동으로 영구 볼륨 회수

14장. 로그 레코드 필드

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

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

최상위 수준 필드는 모든 레코드에 있을 수 있습니다.

15장. MESSAGE

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

데이터 유형	text
예시 값	**HAPPY**

16장. STRUCTURED

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

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

17장. @TIMESTAMP

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

데이터 유형	date
예시 값	**2015-01-24 14:06:05.071000000 Z**

18장. 호스트 이름

이 로그 메시지가 시작된 호스트의 이름입니다. Kubernetes 클러스터에서 이는 kubernetes.host와 동일합니다.

데이터 유형	keyword

19장. IPADDR4

소스 서버의 IPv4 주소입니다. 배열이 될 수 있습니다.

데이터 유형	ip

20장. IPADDR6

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

데이터 유형	ip

21장. LEVEL

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

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

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

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

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

다른 로깅 시스템의 로그 수준 또는 우선 순위를 이전 목록의 가장 가까운 일치 항목에 매핑합니다. 예를

들어 python 로깅에서는 **CRITICAL**은 **crit**로 **ERROR**는 **err** 등과 일치시킬 수 있습니다.

데이터 유형	keyword
예시 값	**info**

22장. PID

사용 가능한 경우 이는 로깅 엔티티의 프로세스 ID입니다.

데이터 유형	keyword

23장. SERVICE

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

데이터 유형	keyword

24장. TAGS

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

데이터 유형	text

25장. FILE

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

데이터 유형	text

26장. OFFSET

오프셋 값입니다. 단일 로그 파일의 컨텍스트에서 값이 엄격하게 단조롭게 증가하는 한 파일에서 로그 라인의 시작까지의 바이트 수(0 또는 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

27장. KUBERNETES

쿠버네티스 관련 메타데이터의 네임스페이스입니다.

데이터 유형	group

27.1. KUBERNETES.POD_NAME

Pod의 이름입니다.

데이터 유형	keyword

27.2. KUBERNETES.POD_ID

Pod의 Kubernetes ID입니다.

데이터 유형	keyword

27.3. KUBERNETES.NAMESPACE_NAME

Kubernetes의 네임스페이스 이름입니다.

데이터 유형	keyword

27.4. KUBERNETES.NAMESPACE_ID

Kubernetes의 네임스페이스 ID입니다.

데이터 유형	keyword

27.5. KUBERNETES.HOST

Kubernetes 노드 이름입니다.

데이터 유형	keyword

27.6. KUBERNETES.CONTAINER_NAME

Kubernetes의 컨테이너 이름입니다.

데이터 유형	keyword

27.7. KUBERNETES.ANNOTATIONS

Kubernetes 오브젝트와 관련된 주석입니다.

데이터 유형	group

27.8. KUBERNETES.LABELS

원래 Kubernetes Pod에 있는 레이블입니다.

데이터 유형	group

27.9. KUBERNETES.EVENT

Kubernetes 마스터 API에서 얻은 Kubernetes 이벤트입니다. 이 이벤트 설명은 Event v1 코어의 type Event를 대략적으로 따릅니다.

데이터 유형	group

27.9.1. kubernetes.event.verb

이벤트 유형, ADDED,MODIFIED 또는 DELETED

데이터 유형	keyword
예시 값	**ADDED**

27.9.2. kubernetes.event.metadata

이벤트 생성 위치 및 시간 관련 정보입니다.

데이터 유형	group

27.9.2.1. kubernetes.event.metadata.name

이벤트 생성을 트리거한 오브젝트의 이름입니다.

데이터 유형	keyword
예시 값	**java-mainclass-1.14d888a4cfc24890**

27.9.2.2. kubernetes.event.metadata.namespace

이벤트가 처음 발생한 네임스페이스의 이름입니다. **eventrouter** 애플리케이션이 배포된 네임스페이스인 **kubernetes.namespace_name**과 다릅니다.

데이터 유형	keyword
예시 값	**default**

27.9.2.3. kubernetes.event.metadata.selfLink

이벤트에 대한 링크입니다.

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

27.9.2.4. kubernetes.event.metadata.uid

이벤트의 고유 ID입니다.

데이터 유형	keyword

예시 값	**d828ac69-7b58-11e7-9cf5-5254002f560c**

27.9.2.5. 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**

27.9.3. kubernetes.event.involvedObject

이벤트의 오브젝트입니다.

데이터 유형	group

27.9.3.1. kubernetes.event.involvedObject.kind

오브젝트 유형입니다.

데이터 유형	keyword
예시 값	**ReplicationController**

27.9.3.2. kubernetes.event.involvedObject.namespace

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

데이터 유형	keyword
예시 값	**default**

27.9.3.3. kubernetes.event.involvedObject.name

이벤트를 트리거한 오브젝트의 이름입니다.

데이터 유형	keyword
예시 값	**java-mainclass-1**

27.9.3.4. kubernetes.event.involvedObject.uid

오브젝트의 고유 ID입니다.

데이터 유형	keyword
예시 값	**e6bff941-76a8-11e7-8193-5254002f560c**

27.9.3.5. kubernetes.event.involvedObject.apiVersion

kubernetes 마스터 API의 버전입니다.

데이터 유형	keyword
예시 값	**v1**

27.9.3.6. kubernetes.event.involvedObject.resourceVersion

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

데이터 유형	keyword
예시 값	**308882**

27.9.4. kubernetes.event.reason

이 이벤트를 생성하는 이유를 제공하는 짧은 머신 이해 문자열입니다.

데이터 유형	keyword

예시 값	**SuccessfulCreate**

27.9.5. kubernetes.event.source_component

이 이벤트를 보고한 구성 요소입니다.

데이터 유형	keyword
예시 값	**replication-controller**

27.9.6. kubernetes.event.firstTimestamp

이벤트가 처음 기록된 시간입니다.

데이터 유형	date
예시 값	**2017-08-07 10:11:57.000000000 Z**

27.9.7. kubernetes.event.count

이 이벤트가 발생한 횟수입니다.

데이터 유형	integer
예시 값	**1**

27.9.8. kubernetes.event.type

이벤트 유형, Normal 또는 Warning입니다. 새 유형을 나중에 추가할 수 있습니다.

데이터 유형	keyword
예시 값	**Normal**

28장. OPENSHIFT

openshift-logging 특정 메타데이터의 네임스페이스

데이터 유형	group

28.1. OPENSHIFT.LABELS

Cluster Log Forwarder 구성에 추가된 레이블

데이터 유형	group