Red Hat OpenShift

Understanding the logging subsystem for Red Hat OpenShift

As a cluster administrator, you can deploy the logging subsystem to aggregate all the logs from your OpenShift Container Platform cluster, such as node system audit logs, application container logs, and infrastructure logs. The logging subsystem aggregates these logs from throughout your cluster and stores them in a default log store. You can use the Kibana web console to visualize log data.

The logging subsystem aggregates the following types of logs:

- `application` - Container logs generated by user applications running in the cluster, except infrastructure container applications.

- `infrastructure` - Logs generated by infrastructure components running in the cluster and OpenShift Container Platform nodes, such as journal logs. Infrastructure components are pods that run in the `openshift*`, `kube*`, or `default` projects.
- `audit` - Logs generated by auditd, the node audit system, which are stored in the **/var/log/audit/audit.log** file, and the audit logs from the Kubernetes apiserver and the OpenShift apiserver.

Because the internal OpenShift Container Platform Elasticsearch log store does not provide secure storage for audit logs, audit logs are not stored in the internal Elasticsearch instance by default. If you want to send the audit logs to the default internal Elasticsearch log store, for example to view the audit logs in Kibana, you must use the Log Forwarding API as described in Forward audit logs to the log store.

Glossary of common terms for OpenShift Container Platform Logging

This glossary defines common terms that are used in the OpenShift Container Platform Logging content.

annotation
You can use annotations to attach metadata to objects.

Cluster Logging Operator (CLO)
The Cluster Logging Operator provides a set of APIs to control the collection and forwarding of application, infrastructure, and audit logs.

Custom Resource (CR)
A CR is an extension of the Kubernetes API. To configure OpenShift Container Platform Logging and log forwarding, you can customize the `ClusterLogging` and the `ClusterLogForwarder` custom resources.

event router

The event router is a pod that watches OpenShift Container Platform events. It collects logs by using OpenShift Container Platform Logging.

Fluentd
Fluentd is a log collector that resides on each OpenShift Container Platform node. It gathers application, infrastructure, and audit logs and forwards them to different outputs.

garbage collection
Garbage collection is the process of cleaning up cluster resources, such as terminated containers and images that are not referenced by any running pods.

Elasticsearch
Elasticsearch is a distributed search and analytics engine. OpenShift Container Platform uses ELasticsearch as a default log store for OpenShift Container Platform Logging.

Elasticsearch Operator
Elasticsearch operator is used to run Elasticsearch cluster on top of OpenShift Container Platform. The Elasticsearch Operator provides self-service for the Elasticsearch cluster operations and is used by OpenShift Container Platform Logging.

indexing
Indexing is a data structure technique that is used to quickly locate and access data. Indexing optimizes the performance by minimizing the amount of disk access required when a query is processed.

JSON logging
OpenShift Container Platform Logging Log Forwarding API enables you to parse JSON logs into a structured object and forward them to either OpenShift Container Platform Logging-managed Elasticsearch or any other third-party system supported by the Log Forwarding API.

Kibana
Kibana is a browser-based console interface to query, discover, and visualize your Elasticsearch data through histograms, line graphs, and pie charts.

Kubernetes API server

Kubernetes API server validates and configures data for the API objects.

Labels
Labels are key-value pairs that you can use to organize and select subsets of objects, such as a pod.

Logging
With OpenShift Container Platform Logging you can aggregate application, infrastructure, and audit logs throughout your cluster. You can also store them to a default log store, forward them to third party systems, and query and visualize the stored logs in the default log store.

logging collector
A logging collector collects logs from the cluster, formats them, and forwards them to the log store or third party systems.

log store
A log store is used to store aggregated logs. You can use the default Elasticsearch log store or forward logs to external log stores. The default log store is optimized and tested for short-term storage.

log visualizer
Log visualizer is the user interface (UI) component you can use to view information such as logs, graphs, charts, and other metrics. The current implementation is Kibana.

node
A node is a worker machine in the OpenShift Container Platform cluster. A node is either a virtual machine (VM) or a physical machine.

Operators
Operators are the preferred method of packaging, deploying, and managing a Kubernetes application in an OpenShift Container Platform cluster. An Operator takes human operational knowledge and encodes it into software that is packaged and shared with customers.

pod
A pod is the smallest logical unit in Kubernetes. A pod consists of one or more containers and runs on a worker node..

Role-based access control (RBAC)
RBAC is a key security control to ensure that cluster users and workloads have access only to resources required to execute their roles.

shards
Elasticsearch organizes the log data from Fluentd into datastores, or indices, then subdivides each index into multiple pieces called shards.

taint
Taints ensure that pods are scheduled onto appropriate nodes. You can apply one or more taints on a node.

toleration
You can apply tolerations to pods. Tolerations allow the scheduler to schedule pods with matching taints.

web console
A user interface (UI) to manage OpenShift Container Platform.

About deploying the logging subsystem for Red Hat OpenShift

OpenShift Container Platform cluster administrators can deploy the logging subsystem using the OpenShift Container Platform web console or CLI to install the OpenShift Elasticsearch Operator and Red Hat OpenShift Logging Operator. When the Operators are installed, you create a `ClusterLogging` custom resource (CR) to schedule logging subsystem pods and other resources necessary to support the logging subsystem. The Operators are responsible for deploying, upgrading, and maintaining the logging subsystem.

The `ClusterLogging` CR defines a complete logging subsystem environment that includes all the components of the logging stack to collect, store and visualize logs. The Red Hat OpenShift Logging Operator watches the logging subsystem CR and adjusts the logging deployment accordingly.

Administrators and application developers can view the logs of the projects for which they have view access.

For information, see Configuring the log collector.

About JSON OpenShift Container Platform Logging

You can use JSON logging to configure the Log Forwarding API to parse JSON strings into a structured object. You can perform the following tasks:

- Parse JSON logs
- Configure JSON log data for Elasticsearch
- Forward JSON logs to the Elasticsearch log store

About collecting and storing Kubernetes events

The OpenShift Container Platform Event Router is a pod that watches Kubernetes events and logs them for collection by OpenShift Container Platform Logging. You must manually deploy the Event Router.

For information, see About collecting and storing Kubernetes events.

About updating OpenShift Container Platform Logging

OpenShift Container Platform allows you to update OpenShift Container Platform logging. You must update the following operators while updating OpenShift Container Platform Logging:

- Elasticsearch Operator
- Cluster Logging Operator

For information, see About updating OpenShift Container Platform Logging.

About viewing the cluster dashboard

The OpenShift Container Platform Logging dashboard contains charts that show details about your Elasticsearch instance at the cluster level. These charts help you diagnose and anticipate problems.

For information, see About viewing the cluster dashboard.

About troubleshooting OpenShift Container Platform Logging

You can troubleshoot the logging issues by performing the following tasks:

- Viewing logging status
- Viewing the status of the log store
- Understanding logging alerts
- Collecting logging data for Red Hat Support
- Troubleshooting for critical alerts

About uninstalling OpenShift Container Platform Logging

You can stop log aggregation by deleting the ClusterLogging custom resource (CR). After deleting the CR, there are other cluster logging components that remain, which you can optionally remove.

For information, see About uninstalling OpenShift Container Platform Logging.

About exporting fields

The logging system exports fields. Exported fields are present in the log records and are available for searching from Elasticsearch and Kibana.

For information, see About exporting fields.

About logging subsystem components

The logging subsystem components include a collector deployed to each node in the OpenShift Container Platform cluster that collects all node and container logs and writes them to a log store. You can use a centralized web UI to create rich visualizations and dashboards with the aggregated data.

The major components of the logging subsystem are:

- collection - This is the component that collects logs from the cluster, formats them, and forwards them to the log store. The current implementation is Fluentd.

- log store - This is where the logs are stored. The default implementation is Elasticsearch. You can use the default Elasticsearch log store or forward logs to external log stores. The default log store is optimized and tested for short-term storage.
- visualization - This is the UI component you can use to view logs, graphs, charts, and so forth. The current implementation is Kibana.

This document might refer to log store or Elasticsearch, visualization or Kibana, collection or Fluentd, interchangeably, except where noted.

About the logging collector

The logging subsystem for Red Hat OpenShift collects container and node logs.

By default, the log collector uses the following sources:

- journald for all system logs
- `/var/log/containers/*.log` for all container logs

If you configure the log collector to collect audit logs, it gets them from `/var/log/audit/audit.log`.

The logging collector is a daemon set that deploys pods to each OpenShift Container Platform node. System and infrastructure logs are generated by journald log messages from the operating system, the container runtime, and OpenShift Container Platform. Application logs are generated by the CRI-O container engine. Fluentd collects the logs from these sources and forwards them internally or externally as you configure in OpenShift Container Platform.

The container runtimes provide minimal information to identify the source of log messages: project, pod name, and container ID. This information is not sufficient to uniquely identify the source of the logs. If a pod with a given name and project is deleted before the log collector begins processing its logs, information from the API server, such as labels and annotations, might not be available. There might not be a way to distinguish the log messages from a similarly named pod and project or trace the logs to their source. This limitation means that log collection and normalization are considered **best effort**.

The available container runtimes provide minimal information to identify the source of log messages and do not guarantee unique individual log messages or that these messages can be traced to their source.

For information, see Configuring the log collector.

About the log store

By default, OpenShift Container Platform uses Elasticsearch (ES) to store log data. Optionally you can use the Log Forwarder API to forward logs to an external store. Several types of store are supported, including fluentd, rsyslog, kafka and others.

The logging subsystem Elasticsearch instance is optimized and tested for short term storage, approximately seven days. If you want to retain your logs over a longer term, it is recommended you move the data to a third-party storage system.

Elasticsearch organizes the log data from Fluentd into datastores, or *indices*, then subdivides each index into multiple pieces called *shards*, which it spreads across a set of Elasticsearch nodes in an Elasticsearch cluster. You can configure Elasticsearch to make copies of the shards, called *replicas*, which Elasticsearch also spreads across the Elasticsearch nodes. The `ClusterLogging` custom resource (CR) allows you to specify how the shards are replicated to provide data redundancy and resilience to failure. You can also specify how long the different types of logs are retained using a retention policy in the `ClusterLogging` CR.

The number of primary shards for the index templates is equal to the number of Elasticsearch data nodes.

The Red Hat OpenShift Logging Operator and companion OpenShift Elasticsearch Operator ensure that each Elasticsearch node is deployed using a unique deployment that includes its own storage volume. You can use a `ClusterLogging` custom resource (CR) to increase the number of Elasticsearch nodes, as needed. See the Elasticsearch documentation for considerations involved in configuring storage.

A highly-available Elasticsearch environment requires at least three Elasticsearch nodes, each on a different host.

Role-based access control (RBAC) applied on the Elasticsearch indices enables the controlled access of the logs to the developers. Administrators can access all logs and developers can access only the logs in their projects.

For information, see Configuring the log store.

About logging visualization

OpenShift Container Platform uses Kibana to display the log data collected by Fluentd and indexed by Elasticsearch.

Kibana is a browser-based console interface to query, discover, and visualize your Elasticsearch data through histograms, line graphs, pie charts, and other visualizations.

For information, see Configuring the log visualizer.

About event routing

The Event Router is a pod that watches OpenShift Container Platform events so they can be collected by the logging subsystem for Red Hat OpenShift. The Event Router collects events from all projects and writes them to `STDOUT` . Fluentd collects those events and forwards them into the OpenShift Container Platform Elasticsearch instance. Elasticsearch indexes the events to the `infra` index.

You must manually deploy the Event Router.

For information, see Collecting and storing Kubernetes events.

About log forwarding

By default, the logging subsystem for Red Hat OpenShift sends logs to the default internal Elasticsearch log store, defined in the `ClusterLogging` custom resource (CR). If you want to forward logs to other log aggregators, you can use the log forwarding features to send logs to specific endpoints within or outside your cluster.

For information, see Forwarding logs to third-party systems.

About Vector

Vector is a log collector offered as an alternative to Fluentd for the logging subsystem.

The following outputs are supported:

- `elasticsearch` . An external Elasticsearch instance. The `elasticsearch` output can use a TLS connection.
- `kafka` . A Kafka broker. The `kafka` output can use an unsecured or TLS connection.
- `loki` . Loki, a horizontally scalable, highly available, multitenant log aggregation system.

Enabling Vector

Vector is not enabled by default. Use the following steps to enable Vector on your OpenShift Container Platform cluster.

Vector does not support FIPS Enabled Clusters.

Prerequisites

- OpenShift Container Platform: 4.11

- Logging subsystem for Red Hat OpenShift: 5.4
- FIPS disabled

Procedure

- Edit the `ClusterLogging` custom resource (CR) in the `openshift-logging` project:

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

- Add a `logging.openshift.io/preview-vector-collector: enabled` annotation to the `ClusterLogging` custom resource (CR).
- Add `vector` as a collection type to the `ClusterLogging` custom resource (CR).

```
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: {}
```

Additional resources

- Vector Documentation

Collector features

Table 1. Log Sources

Feature	Fluentd	Vector
App container logs	✓	✓

Feature	Fluentd	Vector
App-specific routing	✓	✓
App-specific routing by namespace	✓	✓
Infra container logs	✓	✓
Infra journal logs	✓	✓
Kube API audit logs	✓	✓
OpenShift API audit logs	✓	✓
Open Virtual Network (OVN) audit logs	✓	✓

Table 2. Outputs

Feature	Fluentd	Vector
Elasticsearch v5-v7	✓	✓
Fluent forward	✓	
Syslog RFC3164	✓	
Syslog RFC5424	✓	
Kafka	✓	✓
Cloudwatch	✓	✓
Loki	✓	✓

Table 3. Authorization and Authentication

Feature	Fluentd	Vector
Elasticsearch certificates	✓	✓
Elasticsearch username / password	✓	✓
Cloudwatch keys	✓	✓
Cloudwatch STS	✓	
Kafka certificates	✓	✓
Kafka username / password	✓	✓
Kafka SASL	✓	✓
Loki bearer token	✓	✓

Table 4. Normalizations and Transformations

Feature	Fluentd	Vector
Viaq data model - app	✓	✓
Viaq data model - infra	✓	✓
Viaq data model - infra(journal)	✓	✓
Viaq data model - Linux audit	✓	✓
Viaq data model - kube-apiserver audit	✓	✓

Feature	Fluentd	Vector
Viaq data model - OpenShift API audit	✓	✓
Viaq data model - OVN	✓	✓
Loglevel Normalization	✓	✓
JSON parsing	✓	✓
Structured Index	✓	✓
Multiline error detection	✓	
Multicontainer / split indices	✓	✓
Flatten labels	✓	✓
CLF static labels	✓	✓

Table 5. Tuning

Feature	Fluentd	Vector
Fluentd readlinelimit	✓	
Fluentd buffer	✓	
- chunklimitsize	✓	
- totallimitsize	✓	
- overflowaction	✓	
- flushthreadcount	✓	

Feature	Fluentd	Vector
- flushmode	✓	
- flushinterval	✓	
- retrywait	✓	
- retrytype	✓	
- retrymaxinterval	✓	
- retrytimeout	✓	

Table 6. Visibility

Feature	Fluentd	Vector
Metrics	✓	✓
Dashboard	✓	✓
Alerts	✓	

Table 7. Miscellaneous

Feature	Fluentd	Vector
Global proxy support	✓	✓
x86 support	✓	✓
ARM support	✓	✓
PowerPC support	✓	✓

Feature	Fluentd	Vector
IBM Z support	✓	✓
IPv6 support	✓	✓
Log event buffering	✓	
Disconnected Cluster	✓	✓