OpenShift Dedicated 4

Logging

Configuring cluster logging in OpenShift Dedicated 4

Last Updated: 2020-06-18

OpenShift Dedicated 4 Logging

Configuring cluster logging in OpenShift Dedicated 4

Legal Notice

Abstract

This document provides instructions for installing, configuring, and using the cluster logging feature. Cluster logging aggregates logs for a range of OpenShift Dedicated services.

Table of Contents

CHAPTER 1. UNDERSTANDING CLUSTER LOGGING AND OPENSHIFT DEDICATED

As an administrator, you can deploy cluster logging to aggregate logs for a range of OpenShift Dedicated services.

Cluster logging runs on worker nodes. As an administrator, you can monitor resource consumption in the console and via Prometheus and Grafana. Due to the high work load required for logging, more worker nodes may be required for your environment.

Logs in OpenShift Dedicated are retained for seven days before rotation. Logging storage is capped at 600GiB. This is independent of a cluster's allocated base storage.

1.1. CLUSTER LOGGING

OpenShift Dedicated administrators can deploy Cluster Logging and Elasticsearch operators via OperatorHub and configure logging in the **openshift-logging** namespace. Configuring logging will deploy Elasticsearch, Fluentd, and Kibana in the **openshift-logging** namespace. The operators are responsible for deploying, upgrading, and maintaining cluster logging.

You can configure cluster logging by modifying the Cluster Logging Custom Resource (CR), named **instance**. The CR defines a complete cluster logging deployment that includes all the components of the logging stack to collect, store and visualize logs. The Cluster Logging Operator watches the **ClusterLogging** Custom Resource and adjusts the logging deployment accordingly.

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

1.1.1. Cluster logging components

The cluster logging components are based upon Elasticsearch, Fluentd, and Kibana (EFK). The collector, Fluentd, is deployed to each node in the OpenShift Dedicated cluster. It collects all node and container logs and writes them to Elasticsearch (ES). Kibana is the centralized, web UI where users and administrators can create rich visualizations and dashboards with the aggregated data.

There are currently 5 different types of cluster logging components:

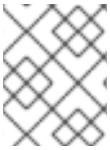
- logStore - This is where the logs will be stored. The current implementation is Elasticsearch.
- collection - This is the component that collects logs from the node, formats them, and stores them in the logStore. The current implementation is Fluentd.
- visualization - This is the UI component used to view logs, graphs, charts, and so forth. The current implementation is Kibana.
- curation - This is the component that trims logs by age. The current implementation is Curator.
- event routing - This is the component forwards OpenShift Dedicated events to cluster logging. The current implementation is Event Router.

In this document, we may refer to logStore or Elasticsearch, visualization or Kibana, curation or Curator, collection or Fluentd, interchangeably, except where noted.

1.1.2. About the logstore

OpenShift Dedicated uses Elasticsearch (ES) to organize the log data from Fluentd into datastores, or *indices*.

Elasticsearch 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*. Elasticsearch also spreads these replicas across the Elasticsearch nodes. The **ClusterLogging** Custom Resource allows you to specify the replication policy in the Custom Resource Definition (CRD) to provide data redundancy and resilience to failure.



NOTE

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

The Cluster Logging Operator and companion Elasticsearch Operator ensure that each Elasticsearch node is deployed using a unique Deployment that includes its own storage volume. You can use a Cluster Logging Custom Resource (CR) to increase the number of Elasticsearch nodes. Refer to Elastic's documentation for considerations involved in choosing storage and network location as directed below.

NOTE

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. Access to the indexes with the **project.{project_name}.{project_uuid}.*** format is restricted based on the permissions of the user in the specific project.

For more information, see Elasticsearch (ES).

1.1.3. About the logging collector

OpenShift Dedicated uses Fluentd to collect data about your cluster.

The logging collector is deployed as a DaemonSet in OpenShift Dedicated that deploys pods to each OpenShift Dedicated node. **journald** is the system log source supplying log messages from the operating system, the container runtime, and OpenShift Dedicated.

The container runtimes provide minimal information to identify the source of log messages: project, pod name, and container id. This 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 log collection and normalization is considered **best effort**.

IMPORTANT

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 more information, see Fluentd.

1.1.4. About logging visualization

OpenShift Dedicated 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, heat maps, built-in geospatial support, and other visualizations.

For more information, see Kibana.

1.1.5. About logging curation

The Elasticsearch Curator tool performs scheduled maintenance operations on a global and/or on a per-project basis. Curator performs actions based on its configuration. Only one Curator Pod is recommended per Elasticsearch cluster.

```
spec:
  curation:
  type: "curator"
  resources:
  curator:
    schedule: "30 3 * * *" (1)
```

(1) Specify the Curator schedule in the cron format.

For more information, see Curator.

1.1.6. About event routing

The Event Router is a pod that forwards OpenShift Dedicated events to cluster logging. You must manually deploy Event Router.

The Event Router collects events and converts them into JSON format, which takes those events and pushes them to **STDOUT**. Fluentd indexes the events to the **.operations** index.

1.1.7. About the Cluster Logging Custom Resource

To make changes to your cluster logging deployment, create and modify the Cluster Logging Custom Resource (CR). Instructions for creating or modifying a CR are provided in this documentation as appropriate.

The following is an example of a typical Custom Resource for cluster logging.

Sample Cluster Logging CR

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

```
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      storage:
        storageClassName: "gp2"
        size: "200Gi"
      redundancyPolicy: "SingleRedundancy"
      nodeSelector:
        node-role.kubernetes.io/worker: ""
      resources:
        request:
          memory: 8G
  visualization:
    type: "kibana"
    kibana:
      replicas: 1
      nodeSelector:
        node-role.kubernetes.io/worker: ""
  curation:
    type: "curator"
    curator:
      schedule: "30 3 * * *"
      nodeSelector:
        node-role.kubernetes.io/worker: ""
  collection:
    logs:
      type: "fluentd"
      fluentd: {}
      nodeSelector:
        node-role.kubernetes.io/worker: ""
```

CHAPTER 2. INSTALLING THE CLUSTER LOGGING AND ELASTICSEARCH OPERATORS

2.1. INSTALLING THE CLUSTER LOGGING AND ELASTICSEARCH OPERATORS

You can use the OpenShift Dedicated console to install cluster logging by deploying instances of the Cluster Logging and Elasticsearch Operators. The Cluster Logging Operator creates and manages the components of the logging stack. The Elasticsearch Operator creates and manages the Elasticsearch cluster used by cluster logging.

NOTE

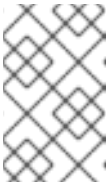
The OpenShift Dedicated cluster logging solution requires that you install both the Cluster Logging Operator and Elasticsearch Operator. When you deploy an instance of the Cluster Logging Operator, it also deploys an instance of the Elasticsearch Operator.

Your OpenShift Dedicated cluster includes 600 GiB of persistent storage that is exclusively available for deploying Elasticsearch for cluster logging.

Elasticsearch is a memory-intensive application. Each Elasticsearch node needs 8G of memory for both memory requests and limits. Each Elasticsearch node can operate with a lower memory setting, though this is not recommended for production deployments.

Procedure

1. Install the Elasticsearch Operator from the OperatorHub:
    a. In the OpenShift Dedicated web console, click **Operators → OperatorHub**.
    b. Choose **Elasticsearch** from the list of available Operators, and click **Install**.
    c. On the **Create Operator Subscription** page, under **A specific namespace on the cluster** select **openshift-logging**. Then, click **Subscribe**.
2. Install the Cluster Logging Operator from the OperatorHub:
    a. In the OpenShift Dedicated web console, click **Operators → OperatorHub**.
    b. Choose **Cluster Logging** from the list of available Operators, and click **Install**.
    c. On the **Create Operator Subscription** page, under **A specific namespace on the cluster** select **openshift-logging**. Then, click **Subscribe**.
3. Verify the operator installations:
    a. Switch to the **Operators → Installed Operators** page.
    b. Ensure that **Cluster Logging** and **Elasticsearch** Operators are listed in the **openshift-logging** project with a **Status** of **InstallSucceeded**.



NOTE

During installation an operator might display a **Failed** status. If the operator then installs with an **InstallSucceeded** message, you can safely ignore the **Failed** message.

If either operator does not appear as installed, to troubleshoot further:

- Switch to the **Operators → Installed Operators** page and inspect the **Status** column for any errors or failures.
- Switch to the **Workloads → Pods** page and check the logs in each Pod in the **openshift-logging** project that is reporting issues.

4. Create and deploy a cluster logging instance:

   a. Switch to the **Operators → Installed Operators** page.

   b. Click the installed **Cluster Logging** Operator.

   c. Under the **Overview** tab, click **Create Instance** . Paste the following YAML definition into the window that displays.

   Cluster Logging Custom Resource (CR)

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: "openshift-logging"
spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      storage:
        storageClassName: "<storage-class-name>"
        size: "200Gi"
      redundancyPolicy: "SingleRedundancy"
      nodeSelector:
        node-role.kubernetes.io/worker: ""
      resources:
        requests:
          memory: 8G
  visualization:
    type: "kibana"
    kibana:
      replicas: 1
      nodeSelector:
        node-role.kubernetes.io/worker: ""
  curation:
    type: "curator"
    curator:
      schedule: "15 * * * *"
      nodeSelector:
```

```
      node-role.kubernetes.io/worker: ""
  collection:
    logs:
      type: "fluentd"
      fluentd: {}
      nodeSelector:
        node-role.kubernetes.io/worker: ""
```

d. Click **Create** to deploy the logging instance, which creates the Cluster Logging and Elasticsearch Custom Resources.

5. Verify that the Pods for the Cluster Logging instance deployed:

   a. Switch to the **Workloads → Pods** page.

   b. Select the **openshift-logging** project.
   You should see several pods for cluster logging, Elasticsearch, Fluentd, and Kibana similar to the following list:

      - 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

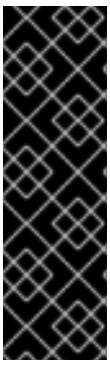
6. Access the Cluster Logging interface, **Kibana**, from the **Monitoring → Logging** page of the OpenShift Dedicated web console.

CHAPTER 3. UPDATING CLUSTER LOGGING

After updating the OpenShift Dedicated cluster from 4.2 to 4.3, you must then upgrade cluster logging from 4.2 to 4.3.

3.1. UPDATING CLUSTER LOGGING

After updating the OpenShift Dedicated cluster, you can update cluster logging from 4.2 to 4.3 by updating the subscription for the Elasticsearch Operator and the Cluster Logging Operator.



IMPORTANT

Changes introduced by the new log forward feature modified the support for **out_forward** starting with the OpenShift Dedicated 4.3 release. In OpenShift Dedicated 4.3, you create a ConfigMap to configure **out_forward**. Any updates to the **secure-forward.conf** section of the Fluentd ConfigMap are removed.

If you use the **out_forward** plug-in, before updating, you can copy your current **secure-forward.conf** section from the Fluentd ConfigMap and use the copied data when you create the **secure-forward** ConfigMap.

Prerequisites

- Update the cluster from 4.2 to 4.3.
- Make sure the cluster logging status is healthy:
  - All Pods are **ready**.
  - Elasticsearch cluster is healthy.
- Optionally, copy your current **secure-forward.conf** section from the Fluentd ConfigMap for use if you want to create the **secure-forward** ConfigMap. See the note above.

Procedure

1. Update the Elasticsearch Operator:
   a. From the web console, click **Operators → Installed Operators**.
   b. Select the **openshift-logging** project.
   c. Click the **Elasticsearch Operator**.
   d. Click **Subscription → Channel**.
   e. In the **Change Subscription Update Channel** window, select **4.3** and click **Save**.
   f. Wait for a few seconds, then click **Operators → Installed Operators**.
   The Elasticsearch Operator is shown as 4.3. For example:

   ```
   Elasticsearch Operator
   4.3.0-201909201915 provided
   by Red Hat, Inc
   ```

2. Update the Cluster Logging Operator:

    a. From the web console, click **Operators → Installed Operators**.

    b. Select the **openshift-logging** Project.

    c. Click the **Cluster Logging Operator**.

    d. Click **Subscription → Channel**.

    e. In the **Change Subscription Update Channel** window, select **4.3** and click **Save**.

    f. Wait for a few seconds, then click **Operators → Installed Operators**.
    The Cluster Logging Operator is shown as 4.3. For example:

    ```
    Cluster Logging
    4.3.0-201909201915 provided
    by Red Hat, Inc
    ```

3. Check the logging components:

    a. Ensure that the Elasticsearch Pods are using a 4.3 image:

    ```
    $ oc get pod -o yaml -n openshift-logging --selector component=elasticsearch |grep
    'image:'

    image: registry.redhat.io/openshift4/ose-logging-elasticsearch5:v4.3.0-202001081344
    image: registry.redhat.io/openshift4/ose-oauth-proxy:v4.3.0-202001081344
    image: registry.redhat.io/openshift4/ose-logging-elasticsearch5:v4.3.0-202001081344
    image: registry.redhat.io/openshift4/ose-oauth-proxy:v4.3.0-202001081344
    image: registry.redhat.io/openshift4/ose-logging-elasticsearch5:v4.3.0-202001081344
    image: registry.redhat.io/openshift4/ose-oauth-proxy:v4.3.0-202001081344
    image: registry.redhat.io/openshift4/ose-logging-elasticsearch5:v4.3.0-202001081344
    image: registry.redhat.io/openshift4/ose-oauth-proxy:v4.3.0-202001081344
    image: registry.redhat.io/openshift4/ose-logging-elasticsearch5:v4.3.0-202001081344
    image: registry.redhat.io/openshift4/ose-oauth-proxy:v4.3.0-202001081344
    image: registry.redhat.io/openshift4/ose-logging-elasticsearch5:v4.3.0-202001081344
    image: registry.redhat.io/openshift4/ose-oauth-proxy:v4.3.0-202001081344
    ```

    b. Ensure that all Elasticsearch Pods are in the **Ready** status:

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

    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
    ```

    c. Ensure that the Elasticsearch cluster is healthy:

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

    {
      "cluster_name" : "elasticsearch",
    ```

```
  "status" : "green",

....
```

d. Ensure that the logging collector Pods are using a 4.3 image:

```
$ oc get pod -n openshift-logging --selector logging-infra=fluentd -o yaml |grep 'image:'

image: registry.redhat.io/openshift4/ose-logging-fluentd:v4.3.0-202001081344
image: registry.redhat.io/openshift4/ose-logging-fluentd:v4.3.0-202001081344
image: registry.redhat.io/openshift4/ose-logging-fluentd:v4.3.0-202001081344
image: registry.redhat.io/openshift4/ose-logging-fluentd:v4.3.0-202001081344
image: registry.redhat.io/openshift4/ose-logging-fluentd:v4.3.0-202001081344
image: registry.redhat.io/openshift4/ose-logging-fluentd:v4.3.0-202001081344
image: registry.redhat.io/openshift4/ose-logging-fluentd:v4.3.0-202001081344
image: registry.redhat.io/openshift4/ose-logging-fluentd:v4.3.0-202001081344
image: registry.redhat.io/openshift4/ose-logging-fluentd:v4.3.0-202001081344
image: registry.redhat.io/openshift4/ose-logging-fluentd:v4.3.0-202001081344
image: registry.redhat.io/openshift4/ose-logging-fluentd:v4.3.0-202001081344
image: registry.redhat.io/openshift4/ose-logging-fluentd:v4.3.0-202001081344
```

e. Ensure that the Kibana Pods are using a 4.3 image:

```
$ oc get pod -n openshift-logging --selector logging-infra=kibana -o yaml |grep 'image:'

image: registry.redhat.io/openshift4/ose-logging-kibana5:v4.3.0-202001081344
image: registry.redhat.io/openshift4/ose-oauth-proxy:v4.3.0-202001081344
image: registry.redhat.io/openshift4/ose-logging-kibana5:v4.3.0-202001081344
image: registry.redhat.io/openshift4/ose-oauth-proxy:v4.3.0-202001081344
```

f. Ensure that the Curator CronJob is using a 4.3 image:

```
$ $ oc get CronJob curator -n openshift-logging -o yaml |grep 'image:'

image: registry.redhat.io/openshift4/ose-logging-curator5:v4.3.0-202001081344
```

CHAPTER 4. VIEWING CLUSTER LOGS USING KIBANA

The cluster logging installation deploys the Kibana web console.

4.1. LAUNCHING KIBANA

Kibana is a browser-based console to query, discover, and visualize your logs through histograms, line graphs, pie charts, heat maps, built-in geospatial support, and other visualizations.

Prerequisites

If you installed OpenShift Dedicated with a proxy, you need to add **.apps.<cluster_name>.<base_domain>** to the **noProxy** list in your cluster-wide Proxy object.

For example:

```
$ oc edit proxy/cluster

apiVersion: config.openshift.io/v1
kind: Proxy
metadata:
  creationTimestamp: "2020-03-30T00:45:44Z"
  generation: 3
  name: cluster
  resourceVersion: "26654"
  selfLink: /apis/config.openshift.io/v1/proxies/cluster
  uid: 2213b41b-0721-4c9f-9586-0678c0058f85
spec:
  httpProxy: http://proxy.com
  httpsProxy: https://proxy.com
  noProxy: .apps.mycluster.example.com 1
  trustedCA:
    name: user-ca-bundle
```

1 Add **.apps.<cluster_name>.<base_domain>** to the **noProxy** list. This is a comma-separated list of destination domain names, domains, IP addresses, or other network CIDRs to exclude proxying.

Procedure

To launch Kibana:

1. In the OpenShift Dedicated console, click **Monitoring → Logging**.
2. Log in using the same credentials you use to log in to the OpenShift Dedicated console. The Kibana interface launches. You can now:
   - Search and browse your data using the Discover page.
   - Chart and map your data using the Visualize page.
   - Create and view custom dashboards using the Dashboard page.
     Use and configuration of the Kibana interface is beyond the scope of this documentation. For more information, on using the interface, see the Kibana documentation.

CHAPTER 5. UNINSTALLING CLUSTER LOGGING

You can remove cluster logging from your OpenShift Dedicated cluster.

5.1. UNINSTALLING CLUSTER LOGGING FROM OPENSHIFT DEDICATED

You can remove cluster logging from your cluster.

Prerequisites

- Cluster logging and Elasticsearch must be installed.

Procedure

To remove cluster logging:

1. Use the following command to remove everything generated during the deployment.

```
$ oc delete clusterlogging instance -n openshift-logging
```

2. Use the following command to remove the Persistent Volume Claims that remain after the Operator instances are deleted:

```
$ oc delete pvc --all -n openshift-logging
```