Jaeger installation, usage, and release notes
**Abstract**

This document provides information on how to use Jaeger in OpenShift Container Platform
<table>
<thead>
<tr>
<th>CHAPTER 1. JAEGER RELEASE NOTES</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1. JAEGER OVERVIEW</td>
<td>3</td>
</tr>
<tr>
<td>1.2. GETTING SUPPORT</td>
<td>3</td>
</tr>
<tr>
<td>1.3. NEW FEATURES OPENSSHIFT JAEGER 1.17.1</td>
<td>3</td>
</tr>
<tr>
<td>1.4. JAEGER KNOWN ISSUES</td>
<td>3</td>
</tr>
<tr>
<td>CHAPTER 2. JAEGER ARCHITECTURE</td>
<td>5</td>
</tr>
<tr>
<td>2.1. JAEGER ARCHITECTURE</td>
<td>5</td>
</tr>
<tr>
<td>2.1.1. Jaeger overview</td>
<td>5</td>
</tr>
<tr>
<td>2.1.2. Jaeger features</td>
<td>5</td>
</tr>
<tr>
<td>2.1.3. Jaeger architecture</td>
<td>6</td>
</tr>
<tr>
<td>CHAPTER 3. JAEGER INSTALLATION</td>
<td>7</td>
</tr>
<tr>
<td>3.1. INSTALLING JAEGER</td>
<td>7</td>
</tr>
<tr>
<td>3.1.1. Jaeger installation overview</td>
<td>7</td>
</tr>
<tr>
<td>3.1.2. Installing the Elasticsearch Operator</td>
<td>7</td>
</tr>
<tr>
<td>3.1.3. Installing the Jaeger Operator</td>
<td>8</td>
</tr>
<tr>
<td>3.2. UPGRADING JAEGER</td>
<td>10</td>
</tr>
<tr>
<td>3.3. REMOVING JAEGER</td>
<td>10</td>
</tr>
<tr>
<td>3.3.1. Removing a Jaeger instance using the web console</td>
<td>10</td>
</tr>
<tr>
<td>3.3.2. Removing a Jaeger instance from the CLI</td>
<td>10</td>
</tr>
<tr>
<td>3.3.3. Removing the Jaeger Operator</td>
<td>11</td>
</tr>
<tr>
<td>CHAPTER 4. INTEGRATING JAEGER</td>
<td>12</td>
</tr>
<tr>
<td>4.1. INTEGRATING JAEGER WITH SERVERLESS APPLICATIONS USING OPENSSHIFT SERVERLESS</td>
<td>12</td>
</tr>
<tr>
<td>4.1.1. Configuring Jaeger for use with OpenShift Serverless</td>
<td>12</td>
</tr>
</tbody>
</table>
CHAPTER 1. JAEGGER RELEASE NOTES

1.1. JAEGGER OVERVIEW

As a service owner, you can use Jaeger to instrument your services to gather insights into your service architecture. Jaeger is an open source distributed tracing platform that you can use for monitoring, network profiling, and troubleshooting the interaction between components in modern, cloud-native, microservices-based applications.

Using Jaeger lets you perform the following functions:

- Monitor distributed transactions
- Optimize performance and latency
- Perform root cause analysis

Jaeger is based on the vendor-neutral OpenTracing APIs and instrumentation.

1.2. GETTING SUPPORT

If you experience difficulty with a procedure described in this documentation, visit the Red Hat Customer Portal. Through the Customer Portal, you can:

- Search or browse through the Red Hat Knowledgebase of technical support articles about Red Hat products.
- Submit a support case to Red Hat Support.

NOTE

When submitting a support case, it is recommended to provide the following information about your cluster to Red Hat Support to aid in troubleshooting:

- Data gathered using the `oc adm must-gather` command
- The unique cluster ID. Navigate to (?) Help → Open Support Case to have the cluster ID autofilled when you submit the case.

- Access other product documentation.

If you have a suggestion for improving this documentation or have found an error, please submit a Bugzilla report against the OpenShift Container Platform product for the Documentation component. Please provide specific details, such as the section name and OpenShift Container Platform version.

1.3. NEW FEATURES OPENSHIFT JAEGGER 1.17.1

This release of OpenShift Jaeger adds support for installing Jaeger as a standalone solution, rather than as a component of Red Hat OpenShift Service Mesh.

1.4. JAEGGER KNOWN ISSUES

These limitations exist in Jaeger:
• While Kafka publisher is included as part of Jaeger, it is not supported.

• Apache Spark is not supported.

• Only self-provisioned Elasticsearch instances are supported. External Elasticsearch instances are not supported in this release.

These are the known issues in Jaeger:

• TRACING-1166 It is not currently possible to use the Jaeger streaming strategy within a disconnected environment. When a Kafka cluster is being provisioned, it results in an error: Failed to pull image registry.redhat.io/amq7/amq-streams-kafka-24-rhel7@sha256:f9ceca004f1b7dccb3b82d9a8027961f9fe4104e0ed69752c0bddd8078b4a1076.

• TRACING-809 Jaeger Ingester is incompatible with Kafka 2.3. When there are two or more instances of the Jaeger Ingester and enough traffic it will continuously generate rebalancing messages in the logs. This is due to a regression in Kafka 2.3 that was fixed in Kafka 2.3.1. For more information, see Jaegertracing-1819.
CHAPTER 2. JAEGER ARCHITECTURE

2.1. JAEGER ARCHITECTURE

Every time a user takes an action in an application, a request is executed by the architecture that may require dozens of different services to participate in order to produce a response. Jaeger lets you perform distributed tracing, which records the path of a request through various microservices that make up an application.

Distributed tracing is a technique that is used to tie the information about different units of work together — usually executed in different processes or hosts — to understand a whole chain of events in a distributed transaction. Developers can visualize call flows in large microservice architectures with distributed tracing. It’s valuable for understanding serialization, parallelism, and sources of latency.

Jaeger records the execution of individual requests across the whole stack of microservices, and presents them as traces. A trace is a data/execution path through the system. An end-to-end trace is comprised of one or more spans.

A span represents a logical unit of work in Jaeger that has an operation name, the start time of the operation, and the duration, as well as potentially tags and logs. Spans may be nested and ordered to model causal relationships.

2.1.1. Jaeger overview

As a service owner, you can use Jaeger to instrument your services to gather insights into your service architecture. Jaeger is an open source distributed tracing platform that you can use for monitoring, network profiling, and troubleshooting the interaction between components in modern, cloud-native, microservices-based applications.

Using Jaeger lets you perform the following functions:

- Monitor distributed transactions
- Optimize performance and latency
- Perform root cause analysis

Jaeger is based on the vendor-neutral OpenTracing APIs and instrumentation.

2.1.2. Jaeger features

Jaeger tracing provides the following capabilities:

- Integration with Kiali – When properly configured, you can view Jaeger data from the Kiali console.
- High scalability – The Jaeger backend is designed to have no single points of failure and to scale with the business needs.
- Distributed Context Propagation – Lets you connect data from different components together to create a complete end-to-end trace.
- Backwards compatibility with Zipkin – Jaeger provides backwards compatibility with Zipkin by accepting spans in Zipkin formats (Thrift or JSON v1/v2) over HTTP.
2.1.3. Jaeger architecture

Jaeger is made up of several components that work together to collect, store, and display tracing data.

- **Jaeger Client** (Tracer, Reporter, instrumented application, client libraries) - Jaeger clients are language specific implementations of the OpenTracing API. They can be used to instrument applications for distributed tracing either manually or with a variety of existing open source frameworks, such as Camel (Fuse), Spring Boot (RHOAR), MicroProfile (RHOAR/Thorntail), Wildfly (EAP), and many more, that are already integrated with OpenTracing.

- **Jaeger Agent** (Server Queue, Processor Workers) - The Jaeger agent is a network daemon that listens for spans sent over User Datagram Protocol (UDP), which it batches and sends to the collector. The agent is meant to be placed on the same host as the instrumented application. This is typically accomplished by having a sidecar in container environments like Kubernetes.

- **Jaeger Collector** (Queue, Workers) - Similar to the Agent, the Collector is able to receive spans and place them in an internal queue for processing. This allows the collector to return immediately to the client/agent instead of waiting for the span to make its way to the storage.

- **Storage** (Data Store) - Collectors require a persistent storage backend. Jaeger has a pluggable mechanism for span storage. Note that for this release, the only supported storage is Elasticsearch.

- **Query** (Query Service) - Query is a service that retrieves traces from storage.

- **Ingester** (Ingester Service) - Jaeger can use Apache Kafka as a buffer between the collector and the actual backing storage (Elasticsearch). Ingester is a service that reads data from Kafka and writes to another storage backend (Elasticsearch).

- **Jaeger Console** - Jaeger provides a user interface that lets you visualize your distributed tracing data. On the Search page, you can find traces and explore details of the spans that make up an individual trace.
CHAPTER 3. JAEGGER INSTALLATION

3.1. INSTALLING JAEGGER

You can install Jaeger on OpenShift Container Platform in either of two ways:

- You can install Jaeger as part of Red Hat OpenShift Service Mesh. Jaeger is included by default in the Service Mesh installation. To install Jaeger as part of a service mesh, follow the Red Hat Service Mesh Installation instructions.

- If you do not want to install a service mesh, you can use the Jaeger Operator to install the Red Hat build of Jaeger by itself. To install Jaeger without a service mesh, use the following instructions.

Prerequisites

Before you can install OpenShift Jaeger, review the installation activities, and ensure that you meet the prerequisites:

- Possess an active OpenShift Container Platform subscription on your Red Hat account. If you do not have a subscription, contact your sales representative for more information.

- Review the OpenShift Container Platform 4.4 overview.

- Install OpenShift Container Platform 4.4.
  - Install OpenShift Container Platform 4.4 on AWS
  - Install OpenShift Container Platform 4.4 on user-provisioned AWS
  - Install OpenShift Container Platform 4.4 on bare metal
  - Install OpenShift Container Platform 4.4 on vSphere

- Install the version of the OpenShift Container Platform command line utility (the oc client tool) that matches your OpenShift Container Platform version and add it to your path.

- An account with the cluster-admin role.

3.1.1. Jaeger installation overview

The steps for installing OpenShift Jaeger are as follows:

- Review the documentation and determine your deployment strategy.

- If your deployment strategy requires persistent storage, install the Elasticsearch Operator via the OperatorHub.

- Install the Jaeger Operator via the OperatorHub.

- Modify the Jaeger YAML file to support your deployment strategy.

- Deploy one or more instances of Jaeger to your OpenShift Container Platform environment.

3.1.2. Installing the Elasticsearch Operator
The default Jaeger deployment uses in-memory storage because it is designed to be installed quickly for those evaluating Jaeger, giving demonstrations, or using Jaeger in a test environment. If you plan to use Jaeger in production, you must install a persistent storage option, in this case, Elasticsearch.

Prerequisites

- Access to the OpenShift Container Platform web console.
- An account with the cluster-admin role.

WARNING

Do not install Community versions of the Operators. Community Operators are not supported.

Procedure

1. Log in to the OpenShift Container Platform web console as a user with the cluster-admin role.


3. Type Elasticsearch into the filter box to locate the Elasticsearch Operator.

4. Click the Elasticsearch Operator provided by Red Hat to display information about the Operator.

5. Click Install.

6. On the Create Operator Subscription page, select All namespaces on the cluster (default). This installs the Operator in the default openshift-operators project and makes the Operator available to all projects in the cluster.

7. Select the Update Channel that matches your OpenShift Container Platform installation. For example, if you are installing on OpenShift Container Platform version 4.4, select the 4.4 update channel.

8. Select the Automatic Approval Strategy.

   NOTE
   
   The Manual approval strategy requires a user with appropriate credentials to approve the Operator install and subscription process.

9. Click Subscribe.

10. On the Installed Operators page, select the openshift-operators project. Wait until you see that the Elasticsearch Operator shows a status of "InstallSucceeded" before continuing.

3.1.3. Installing the Jaeger Operator
To install Jaeger you use the OperatorHub to install the Jaeger Operator.

By default the Operator is installed in the openshift-operators project.

Prerequisites

- Access to the OpenShift Container Platform web console.
- An account with the cluster-admin role.
- If you require persistent storage, you must also install the Elasticsearch Operator before installing the Jaeger Operator.

WARNING

Do not install Community versions of the Operators. Community Operators are not supported.

Procedure

1. Log in to the OpenShift Container Platform web console as a user with the cluster-admin role.
3. Type Jaeger into the filter to locate the Jaeger Operator.
4. Click the Jaeger Operator provided by Red Hat to display information about the Operator.
5. Click Install.
6. On the Create Operator Subscription page, select All namespaces on the cluster (default). This installs the Operator in the default openshift-operators project and makes the Operator available to all projects in the cluster.
7. Select the stable Update Channel. This will automatically update Jaeger as new versions are released. If you select a maintenance channel, for example, 1.17-stable, you will receive bug fixes and security patches for the length of the support cycle for that version.
   - Select an Approval Strategy. You can select Automatic or Manual updates. If you choose Automatic updates for an installed Operator, when a new version of that Operator is available, the Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your Operator without human intervention. If you select Manual updates, when a newer version of an Operator is available, the OLM creates an update request. As a cluster administrator, you must then manually approve that update request to have the Operator updated to the new version.
8. Click Subscribe.
9. On the Subscription Overview page, select the openshift-operators project. Wait until you see that the Jaeger Operator shows a status of "InstallSucceeded" before continuing.
3.2. UPGRADING JAEGER

The Operator Lifecycle Manager (OLM) controls the installation, upgrade, and role-based access control (RBAC) of Operators in a cluster. The OLM runs by default in OpenShift Container Platform. The OLM queries for available Operators as well as upgrades for installed Operators. For more information about how OpenShift Container Platform handled upgrades, refer to the Operator Lifecycle Manager documentation.

The update approach used by the Jaeger Operator upgrades the managed Jaeger instances to the version associated with the Operator. Whenever a new version of the Jaeger Operator is installed, all the Jaeger application instances managed by the Operator will be upgraded to the Operator’s version. For example, if version 1.10 is installed (both Operator and backend components) and the Operator is upgraded to version 1.11, then as soon as the Operator upgrade has completed, the Operator will scan for running Jaeger instances and upgrade them to 1.11 as well.

3.3. REMOVING JAEGER

The steps for removing Jaeger from a OpenShift Container Platform cluster are as follows:

1. Shut down any Jaeger pods.
2. Remove any Jaeger instances.
3. Remove the Jaeger Operator.

3.3.1. Removing a Jaeger instance using the web console

**NOTE**

When deleting an instance that uses the in-memory storage, all data will be permanently lost. Data stored in a persistent storage (such as Elasticsearch) will not be deleted when a Jaeger instance is removed.

**Procedure**

1. Log in to the OpenShift Container Platform web console.
2. Navigate to Operators → Installed Operators.
3. Select the name of the project where the Operators are installed from the Project menu, for example, jaeger-system.
4. Click the Jaeger Operator.
5. Click the Jaeger tab.
6. Click the Options menu next to the instance you want to delete and select Delete Jaeger.
7. In the confirmation message, click Delete.

3.3.2. Removing a Jaeger instance from the CLI
1. Log in to the OpenShift Container Platform CLI.

   $ oc login

2. To display the Jaeger instances run the command:

   oc get deployments -n <jaeger-project>

   The names of operators have the suffix `-operator`. The following example shows two Jaeger Operators and four Jaeger instances:

   oc get deployments -n jaeger-system
   NAME           READY UP-TO-DATE AVAILABLE AGE
   elasticsearch-operator 1/1 1 1 93m
   jaeger-operator 1/1 1 1 49m
   jaeger-test 1/1 1 1 7m23s
   jaeger-test2 1/1 1 1 6m48s
   tracing1 1/1 1 1 7m8s
   tracing2 1/1 1 1 35m

3. To remove an instance of Jaeger, run the command:

   oc delete jaeger <deployment-name> -n <jaeger-project>

   For example,

   oc delete jaeger tracing2 -n jaeger-system

4. To verify the deletion, run `oc get deployment` again:

   oc get deployments -n <jaeger-project>

   For example,

   oc get deployments -n jaeger-system
   NAME           READY UP-TO-DATE AVAILABLE AGE
   elasticsearch-operator 1/1 1 1 94m
   jaeger-operator 1/1 1 1 50m
   jaeger-test 1/1 1 1 8m14s
   jaeger-test2 1/1 1 1 7m39s
   tracing1 1/1 1 1 7m59s

3.3.3. Removing the Jaeger Operator

   **Procedure**

   1. Follow the instructions for Deleting Operators from a cluster.

      - Remove the Jaeger Operator.

      - After the Jaeger Operator has been removed, if appropriate, remove the Elasticsearch Operator.
CHAPTER 4. INTEGRATING JAEGER

4.1. INTEGRATING JAEGER WITH SERVERLESS APPLICATIONS USING OPENSHIFT SERVERLESS

Using Jaeger with OpenShift Serverless allows you to enable distributed tracing for your serverless applications on OpenShift Container Platform.

4.1.1. Configuring Jaeger for use with OpenShift Serverless

Prerequisites

To configure Jaeger for use with OpenShift Serverless, you will need:

- Cluster administrator permissions on an OpenShift Container Platform cluster.
- A current installation of the Jaeger Operator.

Procedure

1. Create and apply the Jaeger custom resource:

   ```
   $ cat <<EOF | oc apply -f -
   apiVersion: jaegertracing.io/v1
   kind: Jaeger
   metadata:
     name: jaeger
     namespace: default
   EOF
   ```

2. Enable tracing for Knative Serving, by editing the `KnativeServing` resource and adding a YAML configuration for tracing.

   **Tracing YAML example**

   ```
   apiVersion: operator.knative.dev/v1alpha1
   kind: KnativeServing
   metadata:
     name: knative-serving
     namespace: knative-serving
   spec:
     config:
       tracing:
         sample-rate: "0.1"
         backend: zipkin
         zipkin-endpoint: http://jaeger-collector.default.svc.cluster.local:9411/api/v2/spans
         debug: "false"
   ```

   **Note:** The `sample-rate` defines sampling probability. Using `sample-rate: "0.1"` means that 1 in 10 traces will be sampled.
2. **backend** must be set to **zipkin**.

3. The **zipkin-endpoint** must point to your **jaeger-collector** service endpoint. To get this endpoint, substitute the namespace where the Jaeger custom resource is applied.

4. Debugging should be set to **false**. Enabling debug mode by setting **debug: "true"** allows all spans to be sent to the server, bypassing sampling.

**Verification steps**

Access the Jaeger web console to see tracing data. You can access the Jaeger web console by using the **jaeger** route.

1. Get the **jaeger** route’s hostname:

   ```
   $ oc get route jaeger
   NAME   HOST/PORT                         PATH   SERVICES       PORT    TERMINATION
   WILDCARD
   jaeger  jaeger-default.apps.example.com          jaeger-query   <all>   reencrypt     None
   ```

2. Open the endpoint address in your browser to view the console.