Deploying the Splunk OpenTelemetry Collector to gather Kubernetes metrics

Where organizations use Kubernetes for container orchestration, monitoring in Kubernetes environments is essential to application performance. Developers use Kubernetes to develop applications using distributed microservices, introducing new challenges not present with traditional monolithic environments. Understanding your microservices environment requires understanding how requests traverse between different layers of the stack and across multiple services. Modern monitoring tools must monitor these interrelated layers while efficiently correlating application and infrastructure behavior to streamline troubleshooting.

OpenTelemetry is a collection of tools, APIs, and SDKs used to instrument, generate, collect, and export telemetry data (metrics, logs, and traces) to help you analyze your application’s performance and behavior. OpenTelemetry is not an observability back-end – that’s where back-end solutions like Splunk, Prometheus, and Jaeger are helpful. These back-end solutions are where your collected application’s telemetry is exported and then reviewed for analysis.

In this article you’ll learn how to deploy the Splunk OpenTelemetry collector to gather Kubernetes metrics to begin analyzing the performance of your Kubernetes workloads.

How application telemetry is collected

To begin collecting your application’s telemetry data and understanding your Kubernetes workloads, you’ll need to deploy the OpenTelemetry Collector. The OpenTelemetry Collector is a vendor-agnostic implementation of how to receive, process and export telemetry data. It removes the need to run, operate, and maintain multiple agents or collectors. Instead, it provides one collector for all your metrics, traces, and logs to help you understand every aspect of how your Kubernetes workloads and applications are performing.

You can use the Splunk Distribution of the OpenTelemetry Collector which uses the open-source OpenTelemetry Collector core as its upstream, along with log collection with FluentD for a more robust experience when using the Splunk Observability Cloud back end for analysis of your Kubernetes workloads.

How the Splunk OpenTelemetry collector for Kubernetes is deployed

The Splunk OpenTelemetry Connector for Kubernetes installs the Splunk OpenTelemetry Collector on your Kubernetes cluster. Deployment of the Splunk OpenTelemetry Connector for Kubernetes is deployed using a Helm chart. Helm Charts are Kubernetes YAML manifests combined into a single package for easy installation of multiple components into your Kubernetes clusters. Once packaged, installing a Helm Chart into your cluster is done through running a single helm install, which simplifies the deployment of containerized applications. You should install helm on your host running your Kubernetes cluster before you begin.
1. To begin the deployment of the Splunk OpenTelemetry Connector for Kubernetes, log into Splunk Observability Cloud. Once logged in, navigate to the hamburger menu on the top left-hand corner and click **Data Setup**.

In the Connect Your Data window, select **Kubernetes** and click **Add Connection**. This takes you to the data setup wizard, which walks you through the various installation requirements.

2. Input your custom settings about the cluster into the connection wizard.
Options include:

- **Access Token** - The token used to authenticate the integration with Splunk.
- **Cluster Name** - The name used to identify the Kubernetes cluster in Splunk Observability Cloud.
- **Provider** - The cloud provider hosting the Kubernetes cluster. Use “other” for local on-premise installations.
- **Distribution** - The Kubernetes distribution type. Use “other” for local on-premise installations.
- **Add Gateway** - Assigns a gateway to run on one node. You should enable this if your cluster is larger than 25 hosts, as a gateway will improve performance in this scenario.

The data setup wizard will show the necessary steps to install the Splunk OpenTelemetry Connector using Helm based on the information about your Kubernetes cluster you have entered.

3. The installation begins by first adding and updating the Helm chart repository. Once the chart repository is updated, use Helm to install the Splunk OpenTelemetry Connector for Kubernetes. Copy the code on each section to complete the installation.

4. To confirm the installation script has been successful, run `kubectl get pods` on your Kubernetes cluster to list all pods in your cluster. The output will show that both the collector agent and collector receiver have been deployed in your cluster.

5. After about 90 seconds, data begins to populate metrics from your cluster onto Splunk Observability Cloud. To verify this is occurring, navigate to the infrastructure dashboard by selecting the hamburger menu and clicking **Infrastructure**.
6. Click on Kubernetes under the Containers section of the dashboard.
7. The dashboard now shows the cluster map with all nodes and pods in your environment.

Now that the Splunk OpenTelemetry Collector is exporting metrics from your Kubernetes cluster to Splunk Observability Cloud, you can use the metrics collected to identify any potential infrastructure issues affecting your Kubernetes workloads, and unlock the ability to collect data from applications that have been instrumented with OpenTelemetry.

Next steps

The content in this article comes from a previously published blog, one of the thousands of Splunk resources available to help users succeed. In addition, these resources might help you understand and implement this guidance:

- Docs: Install and configure Splunk Distribution of OpenTelemetry Collector