



Splunk Open Telemetry Collector for Kubernetes.

Step 1: Prepare the environment

Step 2: Install Kubectl, Kubernetes and HELM

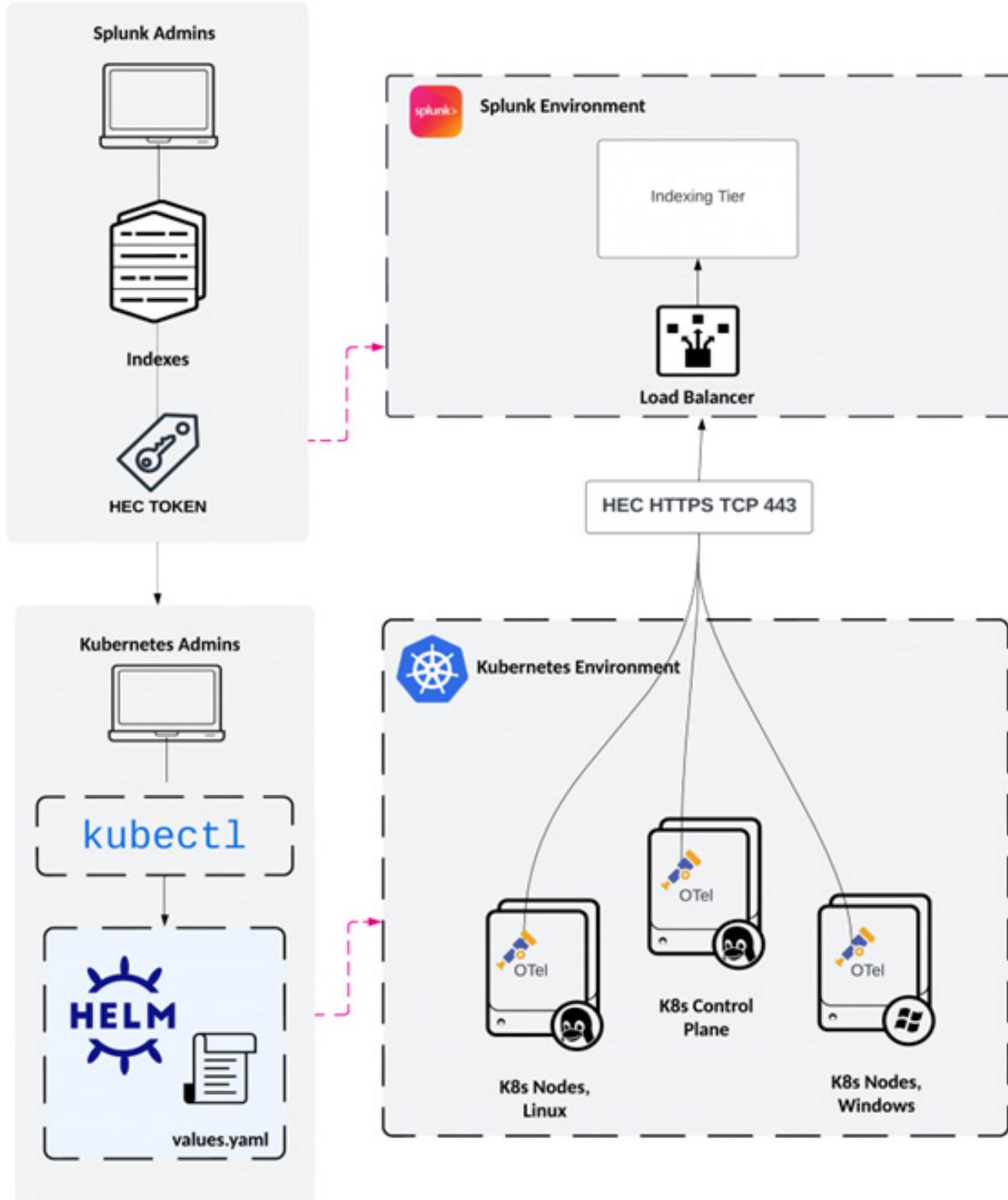
Step 3: Install Otel Collector Helm Chart

1. Pull OTel helm repo locally, by running the following script
2. Open the values file
3. Check your pods for errors, Get running pods
4. View cluster-receiver pod logs with the following command
5. Check Splunk



OpenTelemetry Collector for Kubernetes.

This Splunk validated architecture (SVA) applies to Splunk Cloud Platform and Splunk Enterprise products.



OpenTelemetry Collector incorporates many additional benefits, including:

- Built on OpenTelemetry open standards.
- Improved logging scale with OTel logging versus Fluentd. See <https://github.com/signalfx/splunk-otel-collector-chart/blob/main/docs/advanced-configuration.md#performance-of-native-opentelemetry-logs-collection>
- Advanced metrics collection features.
- Advanced pipeline features including data manipulation.
- Support for trace collection.
- Support for Kubernetes annotations provides ability to route namespace and pod logs to certain indexes, set sourcetypes or even include or exclude logs from being monitored. See <https://github.com/signalfx/splunk-otel-collector-chart/blob/main/docs/advanced-configuration.md#managing-log-ingestion-by-using-annotations>
- Support for multiline logs via the filelog receiver's recombine operator. Users can define their line breaking rules in the collector to ensure multiline logs are properly rendered in Splunk. See : <https://github.com/signalfx/splunk-otel-collector-chart/blob/main/docs/advanced-configuration.md#processing-multi-line-logs>
- Access to OpenTelemetry components. See all components supported in the Splunk Helm chart at <https://github.com/signalfx/splunk-otel-collector/blob/main/docs/components.md#components>

To deploy the OpenTelemetry Collector, some configuration is required by both Splunk admins and Kubernetes admins.

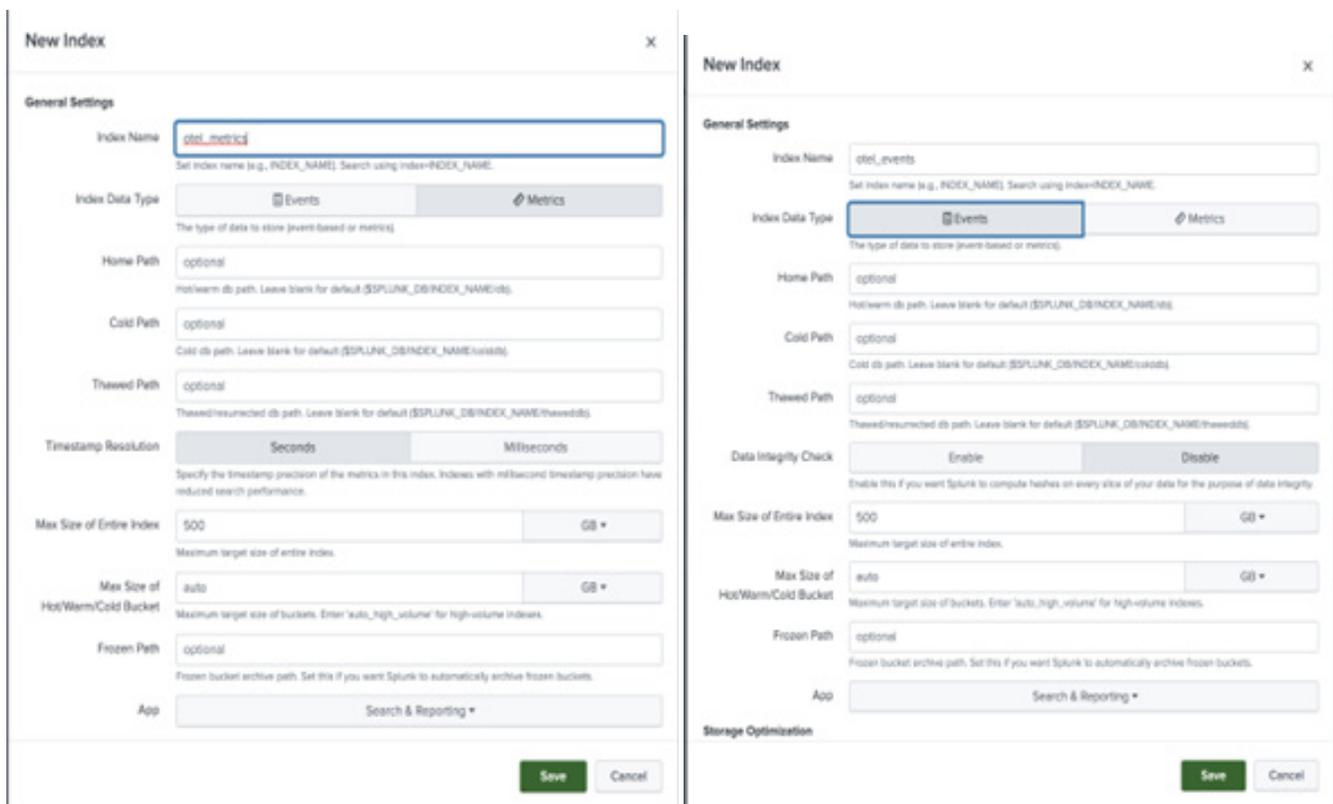
The following sections describe a simple process of getting started with the Splunk OpenTelemetry Collector for Kubernetes (without using the load balancer). Of course a high-level process can be configured depending on the customer's environment and aims in terms of the configuration and use

Step 1: Prepare the enviroment.

1. First you are going to create 1 **events** index called **otel_events** and 1 **metrics** index calles **otel_metrics**

NOTE: YOU ARE BEING ASKED TO CREATE ONE EVENTS INDEX AND ONE METRICS INDEX! PLEASE CONFIGURE THE INDEX DATA TYPE ACCURATELY!!

See below the figures for more instructions on



2. To create the Http Event Collector, go to settings and Data Input, Select HEC Collector, then create a new **HEC Token**

Step 2: Install Kubectl, Kubernetes and HELM

Along with this document, you have been provided with a series of scripts to make your life easier when it comes to deploying all of the technologies that are needed for the lab.

1. Do the following steps locally:

- Download and unzip the files in your local machine and desired directory.
- Give execution permission to the following files:

```
chmod +x microk8s-deploy1.sh microk8s-deploy2.sh otel-deploy.sh
```
- Move the files to the VM where you will be deploying K8S, Kubernetes and HELM. (I do it with scp, but you will know better how to bring them in for your environment).

```
scp microk8s-values.yaml microk8s-deploy1.sh microk8s-deploy2.sh  
otel-deploy.sh<Insert User Here>@<insert IP here>:/home/splunker
```

2. Now, in the VM where you will be deploying the tools, (and counting that the scripts are already there):

```
/microk8s-deploy1.sh
```

3. Then add microk8s to a new group:

```
newgrp microk8s
```

4. Run the second deployment script:

```
./microk8s-deploy2.sh
```

5. Sourcing files:

```
source ~/.bash_aliases
```

Step 3: Install Otel Collector Helm Chart

1. Pull OTelm helm repo locally by running the following script:

```
./otel-deploy.sh
```

This script will run the following commands on your behalf:

```
microk8s helm3 repo add splunk-otel-collector-chart  
https://signalfx.  
github.io/splunk-otel-collector-chart  
microk8s kubectl create ns otel  
microk8s helm3 -n otel install my-splunk-otel-collector -f  
microk8s-values.yaml splunk-otel-collector-chart/splunk-otel-collector
```

2. Open the values file yaml:

```
nano values.yaml
```

Take a minute to look through the different sections contained in the default **values.yaml** file.

Note that the yaml file is sensitive to indentation and may cause parsing issues when deploying the helm chart in the next step.

Find and Update the following values with the created HEC token in Step 1-2 :

```
clusterName: „<yourDesiredName>“
endpoint: „https://<yourSplunkInstance>:8088/services/collector“
token: „00000000-0000-0000-0000-000000000000“ #Your HEC token
goes here
index: „otel_events“
metricsIndex: „otel_metrics“
insecureSkipVerify: true
metricsEnabled: true
logsEngine: otel
containerRuntime: „containerd“
excludeAgentLogs: false
```

When complete, save and quit.

3. Check your pods for errors, Get running pods

```
kubectl -n otel get pods
```

Output should show that both your “agent” and “cluster-receiver” are both running. Be patient, it may take a minute.

```
splunker@so1:~$ kubectl -n otel get pods
NAME                                                                 READY   STATUS    RESTARTS   AGE
mmodestino-tko-otel-splunk-otel-collector-k8s-cluster-rece9h94v   1/1    Running   1 (13h ago) 24h
mmodestino-tko-otel-splunk-otel-collector-agent-ptvmt              1/1    Running   1 (13h ago) 19h
```

You can also use the “watch flag” to keep an eye on the pods as they do their thing...

```
kubectl -n otel get pods -w
```

Press “ctrl + c” to stop the watch.

```
ctrl + c
```

Now, Check both your pods for errors.

```
kubectl -n otel logs -f <collector-agent-pod>
```

Replace <collector-agent-pod> with the agent pod name from previous command.

You should see startup logs showing the collector is running correctly. When running smoothly the pod will not log much info. It should look something like this:

```
2023-02-28T20:48:08.298Z info service/pipelines.go:96 Processor started. {"kind": "processor", "name": "batch", "pipeline": "logs"}
2023-02-28T20:48:08.298Z info service/pipelines.go:92 Processor is starting... {"kind": "processor", "name": "filter/logs", "pipeline": "logs"}
2023-02-28T20:48:08.298Z info service/pipelines.go:96 Processor started. {"kind": "processor", "name": "filter/logs", "pipeline": "logs"}
2023-02-28T20:48:08.298Z info service/pipelines.go:92 Processor is starting... {"kind": "processor", "name": "memory/limits", "pipeline": "logs"}
2023-02-28T20:48:08.298Z info service/pipelines.go:96 Processor started. {"kind": "processor", "name": "memory/limits", "pipeline": "logs"}
2023-02-28T20:48:08.298Z info service/pipelines.go:92 Processor is starting... {"kind": "processor", "name": "resource", "pipeline": "metrics"}
2023-02-28T20:48:08.298Z info service/pipelines.go:96 Processor started. {"kind": "processor", "name": "resource", "pipeline": "metrics"}
2023-02-28T20:48:08.298Z info service/pipelines.go:92 Processor is starting... {"kind": "processor", "name": "resource/detection", "pipeline": "metrics"}
2023-02-28T20:48:08.298Z info service/pipelines.go:96 Processor started. {"kind": "processor", "name": "resource/detection", "pipeline": "metrics"}
2023-02-28T20:48:08.298Z info service/pipelines.go:92 Processor is starting... {"kind": "processor", "name": "batch", "pipeline": "metrics"}
2023-02-28T20:48:08.298Z info service/pipelines.go:96 Processor started. {"kind": "processor", "name": "batch", "pipeline": "metrics"}
2023-02-28T20:48:08.298Z info service/pipelines.go:92 Processor is starting... {"kind": "processor", "name": "memory/limits", "pipeline": "metrics"}
2023-02-28T20:48:08.298Z info service/pipelines.go:96 Processor started. {"kind": "processor", "name": "memory/limits", "pipeline": "metrics"}
2023-02-28T20:48:08.298Z info service/pipelines.go:92 Processor is starting... {"kind": "processor", "name": "resource", "pipeline": "metrics/agent"}
2023-02-28T20:48:08.298Z info service/pipelines.go:96 Processor started. {"kind": "processor", "name": "resource", "pipeline": "metrics/agent"}
2023-02-28T20:48:08.298Z info service/pipelines.go:92 Processor is starting... {"kind": "processor", "name": "resource/detection", "pipeline": "metrics/agent"}
2023-02-28T20:48:08.298Z info service/pipelines.go:96 Processor started. {"kind": "processor", "name": "resource/detection", "pipeline": "metrics/agent"}
2023-02-28T20:48:08.298Z info service/pipelines.go:92 Processor is starting... {"kind": "processor", "name": "resource/add_agent_info", "pipeline": "metrics/agent"}
2023-02-28T20:48:08.298Z info service/pipelines.go:96 Processor started. {"kind": "processor", "name": "resource/add_agent_info", "pipeline": "metrics/agent"}
2023-02-28T20:48:08.298Z info service/pipelines.go:92 Processor is starting... {"kind": "processor", "name": "batch", "pipeline": "metrics/agent"}
2023-02-28T20:48:08.298Z info service/pipelines.go:96 Processor started. {"kind": "processor", "name": "batch", "pipeline": "metrics/agent"}
2023-02-28T20:48:08.298Z info service/pipelines.go:92 Processor is starting... {"kind": "processor", "name": "memory/limits", "pipeline": "metrics/agent"}
2023-02-28T20:48:08.298Z info service/pipelines.go:96 Processor started. {"kind": "processor", "name": "memory/limits", "pipeline": "metrics/agent"}
2023-02-28T20:48:08.298Z info service/pipelines.go:184 Starting receivers...
2023-02-28T20:48:08.297Z info service/pipelines.go:184 Receiver is starting... {"kind": "receiver", "name": "hostmetrics", "pipeline": "metrics"}
2023-02-28T20:48:08.297Z info service/pipelines.go:184 Receiver started. {"kind": "receiver", "name": "hostmetrics", "pipeline": "metrics"}
2023-02-28T20:48:08.297Z info service/pipelines.go:184 Receiver is starting... {"kind": "receiver", "name": "kubernetes", "pipeline": "metrics"}
2023-02-28T20:48:08.297Z info service/pipelines.go:184 Receiver started. {"kind": "receiver", "name": "kubernetes", "pipeline": "metrics"}
2023-02-28T20:48:08.297Z warn internal/warning.go:51 Using the 0.0.0 address exposes this server to every network interface, which may facilitate Denial of Service attacks {"kind": "receiver", "name": "otlp", "pipeline": "metrics", "documentation": "https://github.com/open-telemetry/opentelemetry-collector/blob/main/docs/security-best-practices.md#safeguards-against-denial-of-service-attacks"}
2023-02-28T20:48:08.297Z info otelcolserver/0.47.0/otlp.go:72 Starting gRPC server {"kind": "receiver", "name": "otlp", "pipeline": "metrics", "endpoint": "0.0.0.0:4317"}
2023-02-28T20:48:08.297Z warn internal/warning.go:51 Using the 0.0.0 address exposes this server to every network interface, which may facilitate Denial of Service attacks {"kind": "receiver", "name": "otlp", "pipeline": "metrics", "documentation": "https://github.com/open-telemetry/opentelemetry-collector/blob/main/docs/security-best-practices.md#safeguards-against-denial-of-service-attacks"}
2023-02-28T20:48:08.297Z info otelcolserver/0.47.0/otlp.go:98 Starting HTTP server {"kind": "receiver", "name": "otlp", "pipeline": "metrics", "endpoint": "0.0.0.0:4308"}
2023-02-28T20:48:08.297Z info service/pipelines.go:184 Receiver started. {"kind": "receiver", "name": "otlp", "pipeline": "metrics"}
2023-02-28T20:48:08.297Z info service/pipelines.go:184 Receiver is starting... {"kind": "receiver", "name": "receiver_creator", "pipeline": "metrics"}
2023-02-28T20:48:08.297Z info service/pipelines.go:184 Receiver started. {"kind": "receiver", "name": "receiver_creator", "pipeline": "metrics"}
2023-02-28T20:48:08.297Z warn internal/warning.go:51 Using the 0.0.0 address exposes this server to every network interface, which may facilitate Denial of Service attacks {"kind": "receiver", "name": "signalfn", "pipeline": "metrics", "documentation": "https://github.com/open-telemetry/opentelemetry-collector/blob/main/docs/security-best-practices.md#safeguards-against-denial-of-service-attacks"}
2023-02-28T20:48:08.297Z info service/pipelines.go:184 Receiver started. {"kind": "receiver", "name": "signalfn", "pipeline": "metrics"}
2023-02-28T20:48:08.297Z info service/pipelines.go:184 Receiver is starting... {"kind": "receiver", "name": "prometheus/agent", "pipeline": "metrics"}
2023-02-28T20:48:08.297Z info prometheusreceiver/0.47.0/metrics_receiver.go:143 Setup job added {"kind": "receiver", "name": "prometheus/agent", "pipeline": "metrics", "jobName": "otel-agent"}
2023-02-28T20:48:08.297Z info service/pipelines.go:184 Receiver started. {"kind": "receiver", "name": "prometheus/agent", "pipeline": "metrics"}
2023-02-28T20:48:08.297Z info service/pipelines.go:184 Receiver is starting... {"kind": "receiver", "name": "filelog", "pipeline": "logs"}
2023-02-28T20:48:08.297Z info adapter/receiver.go:54 Starting status receiver {"kind": "receiver", "name": "filelog", "pipeline": "logs"}
2023-02-28T20:48:08.297Z info fileconsumer/file.go:312 Resuming from previously known offset(s). 'start_at' setting is not applicable. {"kind": "receiver", "name": "filelog", "pipeline": "logs", "component": "fileconsumer"}
2023-02-28T20:48:08.298Z info service/pipelines.go:184 Receiver started. {"kind": "receiver", "name": "prometheus/agent", "pipeline": "metrics"}
2023-02-28T20:48:08.298Z info prometheusreceiver/0.47.0/metrics_receiver.go:286 Starting discovery manager {"kind": "receiver", "name": "prometheus/agent", "pipeline": "metrics"}
2023-02-28T20:48:08.298Z info service/pipelines.go:184 Receiver started. {"kind": "receiver", "name": "prometheus/agent", "pipeline": "metrics"}
2023-02-28T20:48:08.298Z info service/pipelines.go:184 Receiver is starting... {"kind": "receiver", "name": "filelog", "pipeline": "logs"}
2023-02-28T20:48:08.298Z info service/pipelines.go:184 Receiver started. {"kind": "receiver", "name": "filelog", "pipeline": "logs"}
2023-02-28T20:48:08.298Z info service/pipelines.go:184 Receiver is starting... {"kind": "receiver", "name": "fluentforward", "pipeline": "logs"}
2023-02-28T20:48:08.298Z info service/pipelines.go:184 Receiver started. {"kind": "receiver", "name": "fluentforward", "pipeline": "logs"}
2023-02-28T20:48:08.298Z info service/pipelines.go:184 Receiver is starting... {"kind": "receiver", "name": "otlp", "pipeline": "logs"}
2023-02-28T20:48:08.298Z info healthcheck/handler.go:129 Health Check state change {"kind": "extension", "name": "health_check", "status": "ready"}
2023-02-28T20:48:08.298Z info service/service.go:185 Everything is ready. Begin running and processing data.
2023-02-28T20:48:08.441Z info fileconsumer/file.go:139 Started watching file {"kind": "receiver", "name": "filelog", "pipeline": "logs", "component": "fileconsumer", "path": "/var/log/pods/kube-system_calico-kube-control-plane-6460378e-4ee2_4916db-899c-4593-111e22a832/calico-sub-controllers&.log"}
2023-02-28T20:48:09.298Z info receivercreator/0.47.0/observer/handler.go:84 starting receiver {"kind": "receiver", "name": "receiver_creator", "pipeline": "metrics", "name": "smrtaagent/credns", "endpoint": "0.0.0.0:4318", "endpoint_id": "tda_observer/04P926-f48e-4243-983-282ab6ac05"}
```

Press “ctrl+c” to exit followtail view

4. View cluster-receiver pod logs with the following command

```
kubectl -n otel logs -f <collector-cluster-reciever-pod>
```

press “ctrl” + “c” to exit followtail view

If there is an issue with the configuration, you will see a constant stream of warnings or errors. You will need to go back and fix your configuration then use the following command to deploy your changes:

```
helm -n otel upgrade <yourDesiredName> -f values.yaml splunk-otel-collector-chart/splunk-otel-collector
```

Once you deploy your config updates, return to the previous step to check your pods are running and there are no errors in your pod logs.

5. Check Splunk

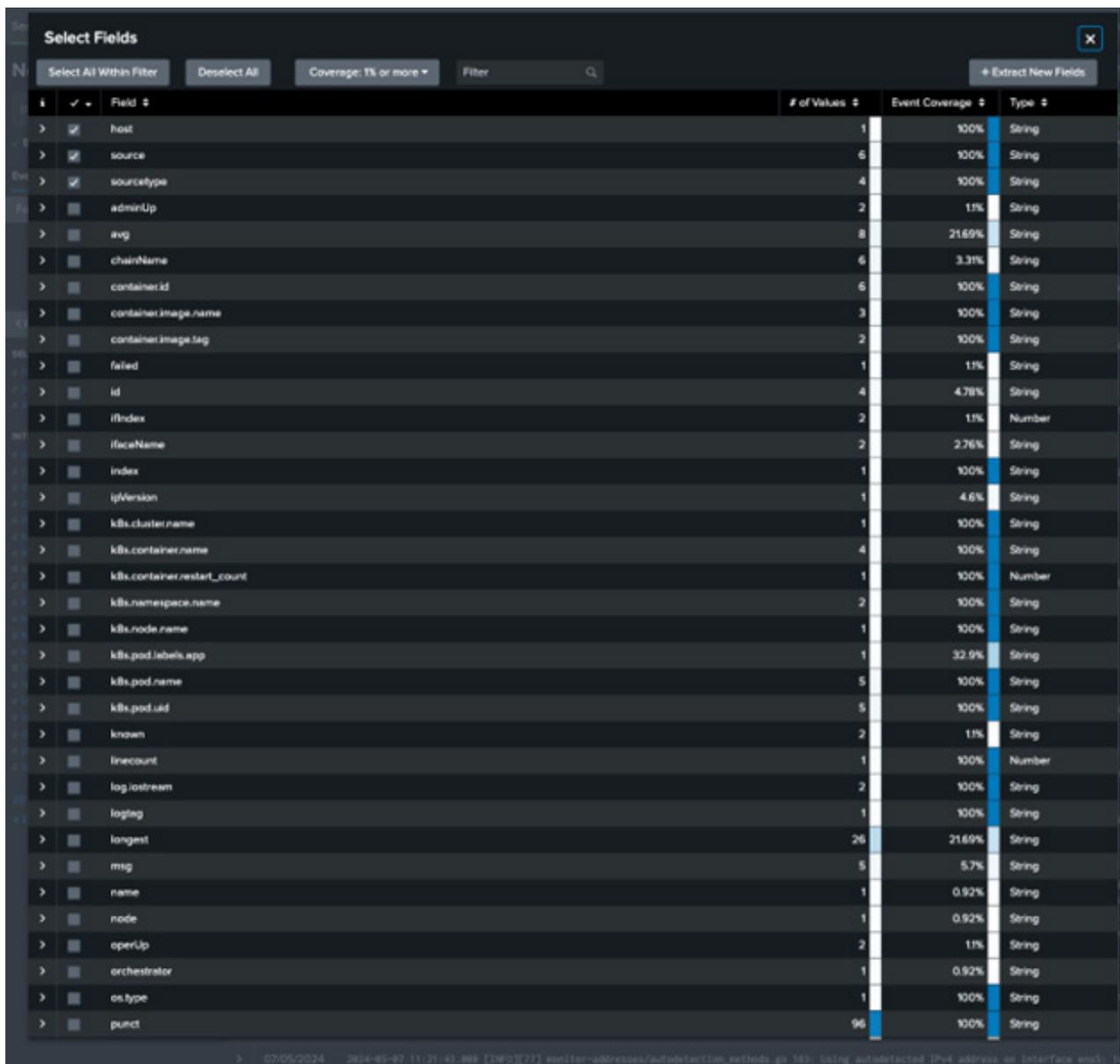
Navigate back to Splunk Web and the Search & Reporting app. Use the default search time range to ensure you are seeing events.

```
index=otel_events
```

The screenshot displays the Splunk Search & Reporting interface. At the top, the search bar contains the query `index=otel_events`. The search results show 544 events from 06/05/2024 11:00:00.000 to 07/05/2024 11:26:54.000. The interface includes a navigation menu with options like Search, Analytics, Datasets, Reports, Alerts, and Dashboards. Below the search bar, there are controls for saving, creating table views, and zooming. The main area shows a list of events with columns for Time and Event. The events are filtered by source type `kube.container:calico-node`. The interface also features a sidebar with field selection options and a visualization area at the top right.

Time	Event
07/05/2024 11:26:43.025	2024-05-07 11:26:43.824 [INFO][77] monitor-addresses/autodetection_methods.go 183: Using autodetected IPv4 address on interface ens1: 10.243.0.21
07/05/2024 11:26:32.689	2024-05-07 11:26:32.688 [INFO][76] felix/summary.go 100: Summarising 11 dataplane reconciliation loops over 1m2.8s: avg=6ms longest=22ms ()
07/05/2024 11:25:43.024	2024-05-07 11:25:43.823 [INFO][77] monitor-addresses/autodetection_methods.go 183: Using autodetected IPv4 address on interface ens1: 10.243.0.21
07/05/2024 11:25:29.923	2024-05-07 11:25:29.922 [INFO][76] felix/summary.go 100: Summarising 8 dataplane reconciliation loops over 1m3.5s: avg=7ms longest=27ms ()
07/05/2024 11:24:43.013	2024-05-07 11:24:43.812 [INFO][77] monitor-addresses/autodetection_methods.go 183: Using autodetected IPv4 address on interface ens1: 10.243.0.21
07/05/2024 11:24:26.379	2024-05-07 11:24:26.378 [INFO][76] felix/summary.go 100: Summarising 18 dataplane reconciliation loops over 1m2.9s: avg=4ms longest=9ms (r esync-nat-v4)
07/05/2024 11:23:43.015	2024-05-07 11:23:43.812 [INFO][77] monitor-addresses/autodetection_methods.go 183: Using autodetected IPv4 address on interface ens1: 10.243.0.21
07/05/2024 11:23:23.522	2024-05-07 11:23:23.521 [INFO][76] felix/summary.go 100: Summarising 13 dataplane reconciliation loops over 1m3s: avg=6ms longest=22ms ()
07/05/2024 11:22:43.001	2024-05-07 11:22:43.801 [INFO][77] monitor-addresses/autodetection_methods.go 183: Using autodetected IPv4 address on interface ens1: 10.243.0.21

If you click on more fields, you'll see:

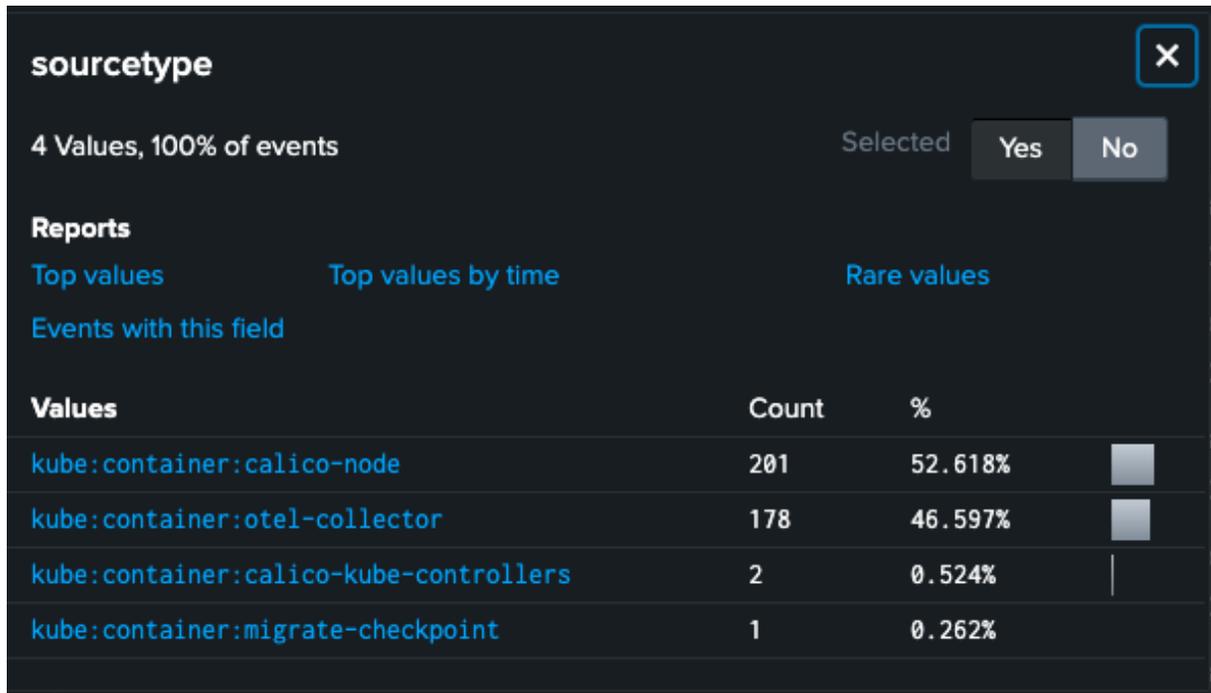


The screenshot shows a 'Select Fields' dialog box with a table of fields. The table has columns for 'Field', '# of Values', 'Event Coverage', and 'Type'. The 'Field' column includes a checkbox and a dropdown arrow. The 'Type' column shows the data type for each field. The total number of fields is 96, and the total event coverage is 100%.

Field	# of Values	Event Coverage	Type
host	1	100%	String
source	6	100%	String
sourcetype	4	100%	String
adminUp	2	1%	String
avg	8	21.69%	String
chainName	6	3.31%	String
container.id	6	100%	String
container.image.name	3	100%	String
container.image.tag	2	100%	String
failed	1	1%	String
id	4	4.78%	String
ifaceIndex	2	1%	Number
ifaceName	2	2.76%	String
index	1	100%	String
ipVersion	1	4.6%	String
k8s.cluster.name	1	100%	String
k8s.container.name	4	100%	String
k8s.container.restart_count	1	100%	Number
k8s.namespace.name	2	100%	String
k8s.node.name	1	100%	String
k8s.pod.labels.app	1	32.9%	String
k8s.pod.name	5	100%	String
k8s.pod.uid	5	100%	String
known	2	1%	String
linecount	1	100%	Number
log.iostream	2	100%	String
logging	1	100%	String
longest	26	21.69%	String
msg	5	5.7%	String
name	1	0.92%	String
node	1	0.92%	String
operUp	2	1%	String
orchestrator	1	0.92%	String
os.type	1	100%	String
punct	96	100%	String

07/05/2024 20:24:45-47:11:21:41:889 [INFO] [77] eventer-addresses/outdetector_methods.go:183: using autodetected IPid address as interface name: 1

The sourcetypes are:



The screenshot shows a configuration window for 'sourcetype'. It indicates '4 Values, 100% of events' and has 'Selected' buttons for 'Yes' and 'No'. Under the 'Reports' section, there are links for 'Top values', 'Top values by time', and 'Rare values', along with 'Events with this field'. A table displays the following data:

Values	Count	%
kube:container:calico-node	201	52.618%
kube:container:otel-collector	178	46.597%
kube:container:calico-kube-controllers	2	0.524%
kube:container:migrate-checkpoint	1	0.262%

The type of metrics that can be displayed using the query:

```
mcatalog values(metric_name) WHERE index=otel_metrics
```

```
system.paging.faults  
system.paging.operations  
system.processes.count  
system.processes.created  
up
```

New Search

```
|mcatalog values(metric_name) WHERE index=otel_metrics
```

✓ 5,494 events (06/05/2024 09:00:00.000 to 07/05/2024 09:31:04.000)

No Event Sampling ▾

Events Patterns **Statistics (1)** Visualization

100 Per Page ▾

↗ Format

Preview ▾

values(metric_name) ⇅

```
container.cpu.time
container.cpu.utilization
container.filesystem.available
container.filesystem.capacity
container.filesystem.usage
container.memory.available
container.memory.major_page_faults
container.memory.page_faults
container.memory.rss
container.memory.usage
container.memory.working_set
coredns.cache_entries
coredns_dns_request_size_bytes
coredns_dns_requests_total
coredns_dns_responses_total
k8s.container.cpu_limit
k8s.container.cpu_request
k8s.container.memory_limit
k8s.container.memory_request
k8s.container.ready
k8s.container.restarts
k8s.daemonset.current_scheduled_nodes
k8s.daemonset.desired_scheduled_nodes
k8s.daemonset.misscheduled_nodes
k8s.daemonset.ready_nodes
k8s.deployment.available
k8s.deployment.desired
k8s.namespace.phase
k8s.node.condition_ready
k8s.node.cpu.time
k8s.node.cpu.utilization
k8s.node.filesystem.available
k8s.node.filesystem.capacity
k8s.node.filesystem.usage
k8s.node.memory.available
k8s.node.memory.major_page_faults
k8s.node.memory.page_faults
k8s.node.memory.rss
k8s.node.memory.usage
k8s.node.memory.working_set
k8s.pod.cpu.time
k8s.pod.cpu.utilization
k8s.pod.filesystem.available
k8s.pod.filesystem.capacity
k8s.pod.filesystem.usage
k8s.pod.memory.available
k8s.pod.memory.major_page_faults
k8s.pod.memory.page_faults
k8s.pod.memory.rss
k8s.pod.memory.usage
```

```
k8s.pod.network.errors
k8s.pod.network.io
k8s.pod.phase
k8s.replicaset.available
k8s.replicaset.desired
otelcol_exporter_queue_capacity
otelcol_exporter_queue_size
otelcol_exporter_send_failed_log_records
otelcol_exporter_send_failed_metric_points
otelcol_exporter_sent_log_records
otelcol_exporter_sent_metric_points
otelcol_otelsvc_k8s_ip_lookup_miss
otelcol_otelsvc_k8s_namespace_added
otelcol_otelsvc_k8s_namespace_updated
otelcol_otelsvc_k8s_pod_added
otelcol_otelsvc_k8s_pod_deleted
otelcol_otelsvc_k8s_pod_table_size
otelcol_otelsvc_k8s_pod_updated
otelcol_process_cpu_seconds
otelcol_process_memory_rss
otelcol_process_runtime_heap_alloc_bytes
otelcol_process_runtime_total_alloc_bytes
otelcol_process_runtime_total_sys_memory_bytes
otelcol_process_uptime
otelcol_processor_accepted_log_records
otelcol_processor_accepted_metric_points
otelcol_processor_dropped_log_records
otelcol_processor_dropped_metric_points
otelcol_processor_filter_logs_filtered
otelcol_processor_refused_log_records
otelcol_processor_refused_metric_points
otelcol_receiver_accepted_log_records
otelcol_receiver_accepted_metric_points
otelcol_receiver_refused_log_records
otelcol_receiver_refused_metric_points
otelcol_scraper_errored_metric_points
otelcol_scraper_scraped_metric_points
scrape_duration_seconds
scrape_samples_post_metric_relabeling
scrape_samples_scraped
scrape_series_added
system.cpu.load_average.15m
system.cpu.load_average.1m
system.cpu.load_average.5m
system.cpu.time
```

Navigate to the Analytics tab and confirm you see metrics, you can check there the different dashboards, set analysis filters, and find the data that you want to analyze

