



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Operations

Sidecar processed logging

Contents

- 1 What does a sidecar container with a logging agent (like Fluent Bit) require?
 - 1.1 Services support for Sidecar logging

Learn about the Sidecar processing of the structured logging to Stdout/Stderr that is available as an option for private edition services.

Related documentation:

•

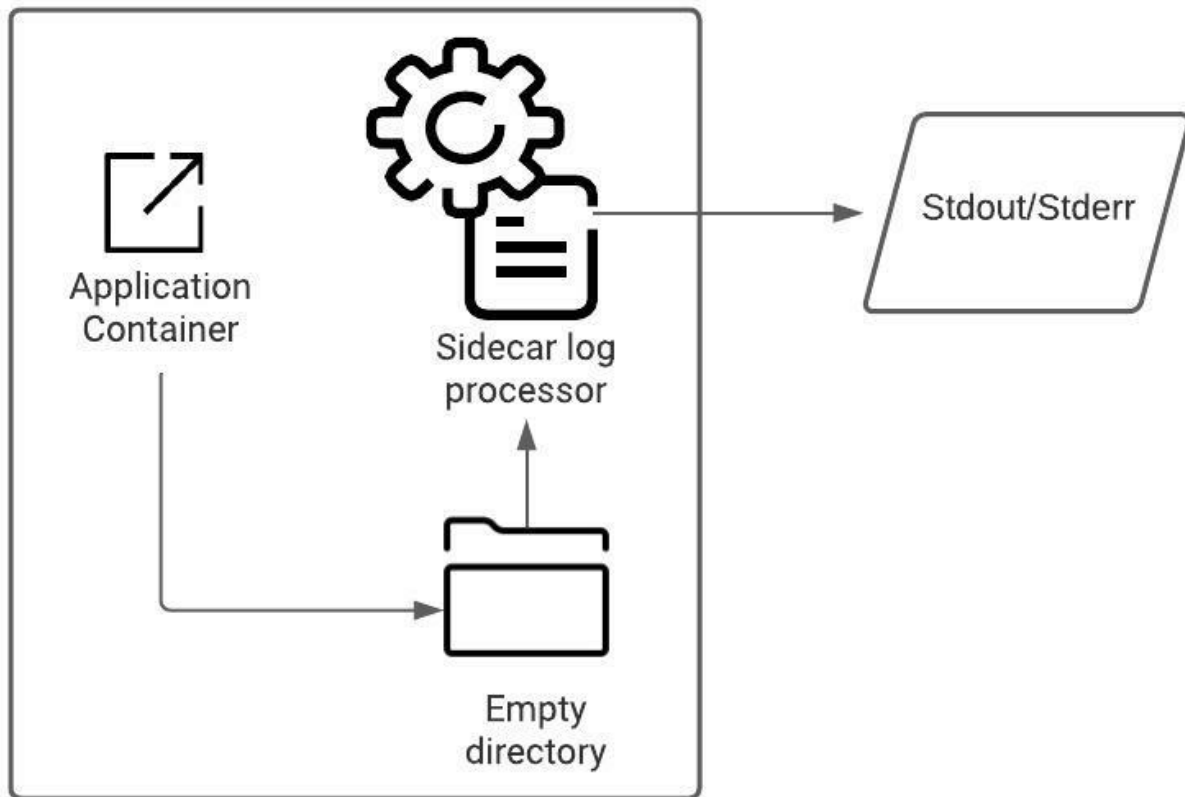
RSS:

- [For private edition](#)

Some Genesys service containers write logs to log files. This method is similar to that of the structured logging in terms of the the log aggregation. Here, a sidecar to be applied to a sidecar container that is applied to the service. The sidecar container processes this data and sends it to stdout/stderr. Any log aggregator (such as Fluentd) picks up this data and applies the same operations as that of standard structured logs.

Services that can log on to stdout/stderr can be ingested into Elasticsearch by using sidecar container for processing the logs. The service writes the logs to **EmptyDir** and sidecar container collects and processes the output to the **/var/log/pods** directory.

A log aggregator will scrape directory and post log data to the Elasticsearch index.



What does a sidecar container with a logging agent (like Fluent Bit) require?

You require a ConfigMap that contains the configuration to configure Fluent Bit. For more information on configuring Fluent Bit, refer to [Fluent Bit Documentation](#).

Here is a ConfigMap sample with Fluent Bit version 1.8.x:

```
## FluentBit Configmap
apiVersion: v1
kind: ConfigMap
metadata:
  name: fluent-bit-config
  labels:
    k8s-app: fluent-bit
data:
  fluent-bit.conf: |
    [SERVICE]
      Flush          1
      Log_Level      debug
      Daemon         off
      Parsers_File   parsers.conf
      HTTP_Server    on
      HTTP_Listen    0.0.0.0
      HTTP_Port      2020
```

```
@INCLUDE input-kubernetes.conf
@INCLUDE output-stdout.conf
input-kubernetes.conf: |
[INPUT]
    Name          tail
    Tag           kube,*
    Path
    Parser        docker
    DB            /var/log/flb_kube.db
    Mem_Buf_Limit 5MB
    Skip_Long_Lines 0n
    Refresh_Interval 10

output-stdout.conf: |
[OUTPUT]
    Name          stdout
    Match         *
```

You also require a pod that has a sidecar container running Fluentd. The pod mounts a volume where Fluentd can pick up its configuration data. Here is an example:

```
volumeMounts:
- name:
  mountPath:
...
volumes:
- name:
  configMap:
    name:
...
image: 'fluent/fluent-bit:'
```

Services support for Sidecar logging

These services have the option use the Sidecar processed logging approach:

- Genesys Customer Experience Insights – GCXI
- Genesys Voice Platform – GVP
- Voice Microservice
- Voice Tenant Service
- Web Based Real-Time Reporting (Pulse)
- WebRTC Media Service
- Gplus WFM