



KubeCon



CloudNativeCon

Europe 2022

WELCOME TO VALENCIA





KubeCon



CloudNativeCon

Europe 2022

About myself

Yaml

```
1 apiVersion: feynmanzhou.github.io/v1alpha1
2 kind: introduction
3 metadata:
4   name: about
5   annotations:
6     feynmanzhou.github.io/full-name: Feynman Zhou (周鹏飞)
7     feynmanzhou.github.io/organization: KubeSphere (https://kubesphere.io)
8     feynmanzhou.github.io/career: Senior Community Manager
9     feynmanzhou.github.io/community: CNCF Ambassador, CDF Ambassador, Fluent Member, InfoQ DevOps Editor
10    feynmanzhou.github.io/company: QingCloud
11    feynmanzhou.github.io/website: https://feynmanzhou.github.io
12
13 spec:
14   skills:
15     - Kubernetes, Linux, Fluent Bit, DevOps, Serverless/FaaS
16     - Technical Writing, Advocacy and Outreach, Host Events
```



PromCon

Feynman Zhou

Community Manager
Open Source Citizen

Build a Cloud Native Logging Pipeline on the Edge with Fluent Operator

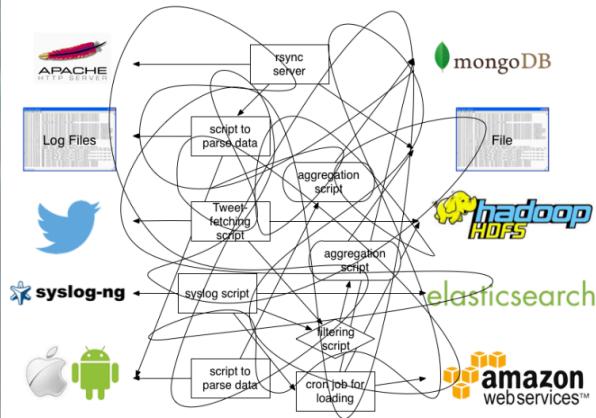
Feynman Zhou, KubeSphere Member, Fluent
Member, CNCF Ambassador



Agenda

- The Challenges of Logging in Kubernetes
- Introduction and comparison to Fluent Bit and Fluentd
- How Fluent Operator empowers Fluent Bit and Fluentd
- Fluent Operator architecture and workflow
- Demo: Using Fluent Operator in K3s to process logs
- Use case: Fluent Bit Operator in the KubeSphere Logging System

The Challenges of Logging in Kubernetes



The Challenges of Logging in Kubernetes



- Logs come from different places
 - BM, VM, Embedded Devices, Edge, Container, Pod, TCP, UDP...
- Different data formats
 - Formats: JSON, Apache, Nginx; Docker, CRI
- Multiple outputs and destinations
 - ES, Loki, Splunk, MongoDB, S3...
- Need to collect and process logs in a performant & efficient way
- Isolation: Data security and reliability in a multi-tenant environment

How to debug your workloads in Kubernetes



KubeCon



CloudNativeCon

Europe 2022

- Native method -

```
$ kubectl logs pod1 -c container2
```

- Cloud providers' logging solutions: Stackdriver, Cloudwatch...
- ISVs' logging solution: Splunk, Sumo Logic, Datadog
- OSS logging solutions: ELK, Loki, **Fluent Bit**, **Fluentd**...

Fluent Bit is a Swiss Knife for logs processing, forwarding and distribution

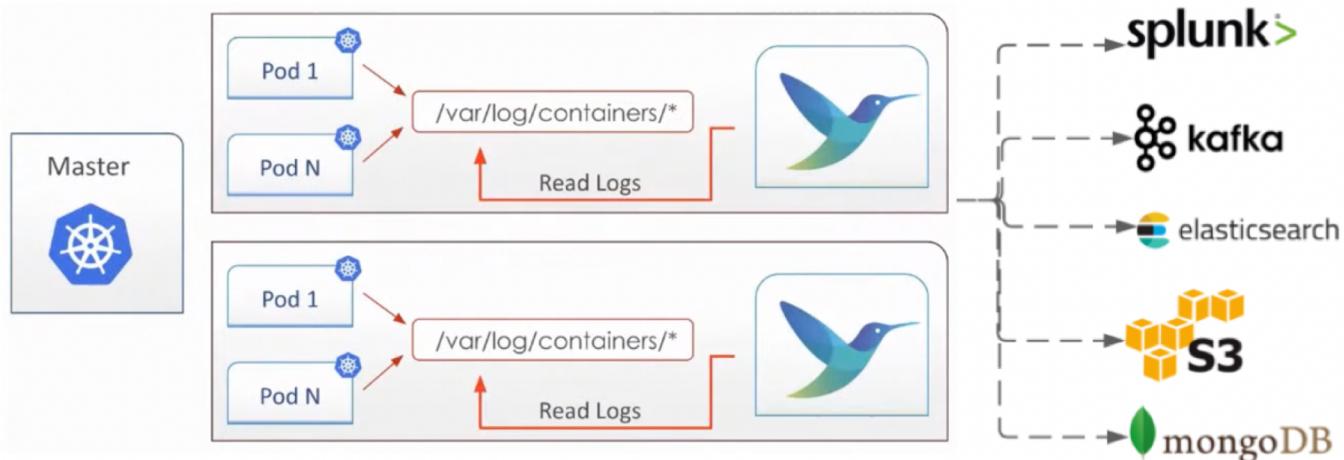


A light-weight & high-performance log processor

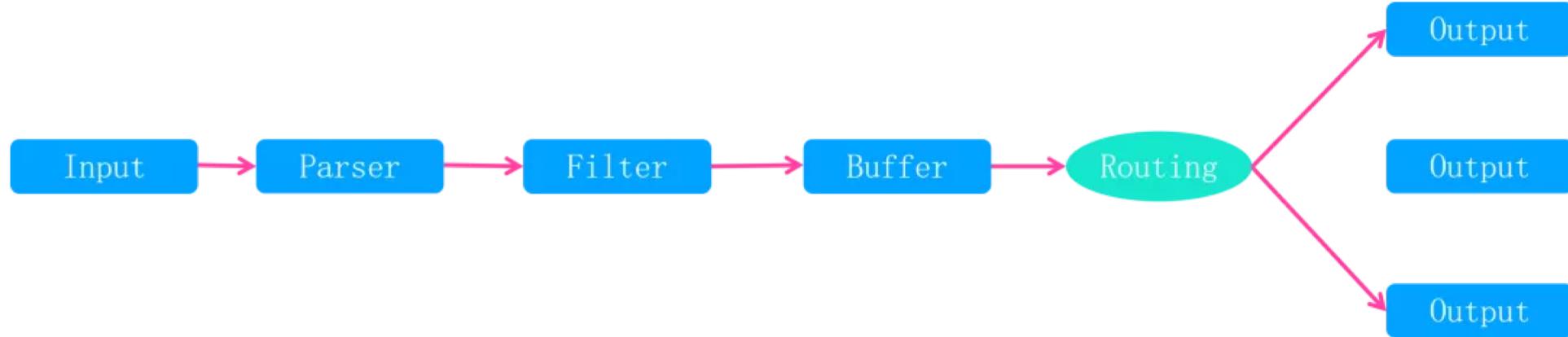
Introduction to Fluent Bit and Fluentd

Fluent Bit

- Started in 2015, a CNCF sub-project under the umbrella of Fluentd
- Written in C, lightweight and zero dependencies
- Pluggable, around 70 plugins available
- high throughput with low CPU and Memory usage (~640KB)



Fluent Bit Logging Pipeline and Plugins



Input
Metrics Forward HTTP Stdin Systemd Tail
Gather information from different sources

Parser
JSON Regular Expression Logfmt Decoders
Convert from unstructured to structured data.

Filter
Except Grep Lua Modify Nest Parser
Record Modifier Kubernetes
To match, exclude or enrich logs with some specific metadata

Output
File Forward HTTP Kafka Loki ES
Define destinations for the data, e.g. remote services, local FS



KubeCon



CloudNativeCon

Europe 2022

**Fluentd is a data collector, which lets you unify
the data collection and consumption for a
better use and understanding of data.**

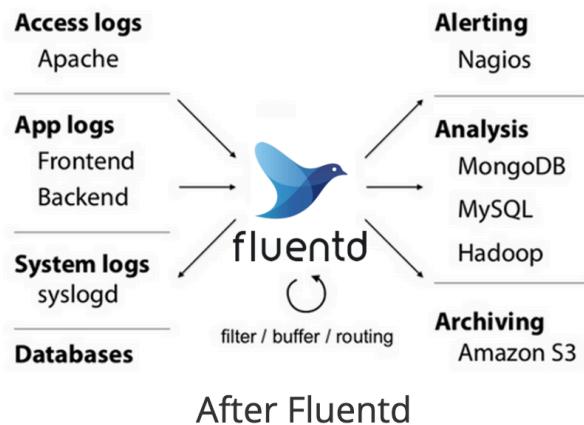
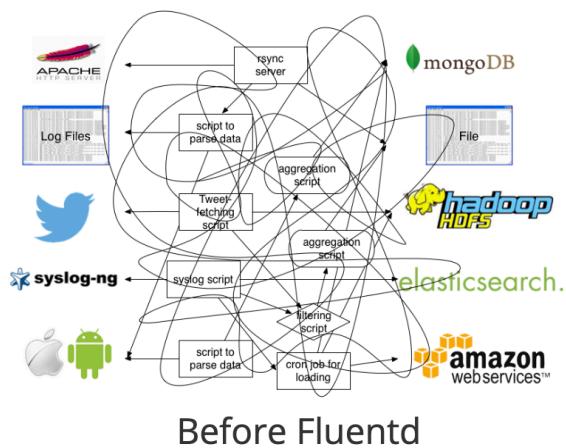


fluentd Build Your Unified Logging Layer

Introduction to Fluent Bit and Fluentd

Fluentd

- Started in 2011, a CNCF graduated project
- Written in C & Ruby. Built as a Ruby Gem, it requires a certain number of gems.
- Pluggable, around 1000 plugins available
- High Performance but also consumes high memory (~40MB)



Fluent Bit vs Fluentd

Text	Fluentd	Fluent Bit
Scope	Containers / Servers	Embedded Linux / Containers / Servers
Language	C & Ruby	C
Memory	~40MB	~650KB
Performance	High Performance	High Performance
Dependencies	Built as a Ruby Gem, it requires a certain number of gems.	Zero dependencies, unless some special plugin requires them.
Plugins	More than 1000 plugins available	Around 70 plugins available
License	Apache License v2.0	Apache License v2.0

Fluent Operator: Facilitate the Management of Fluent Bit & Fluentd

Fluent Operator Introduction - History



- Open sourced as FluentBit Operator by KubeSphere in Jan, 2019
 - FluentBit Operator v0.1.0 ~ v0.8.0
- Donated to Fluent community in Aug, 2021
 - FluentBit Operator v0.9.0 ~ v0.13.0
- Renamed to Fluent Operator in Mar, 2022
 - Add Fluentd support
 - Change FluentBit CRDs from namespaced to cluster scope
 - Fluent Operator v1.0.0

Fluent Bit can not reload config gracefully



KubeCon

CloudNativeCon

Europe 2022

[fluent / fluent-bit](#)

Code Issues 275 Pull requests 34 Projects 0 Wiki Security Insights

Restart when config directory has changes #842

Closed donbowman wants to merge 7 commits into [fluent:master](#) from [donbowman:restart-config-changed](#)

Conversation 7 Commits 7 Checks 1 Files changed 7

donbowman commented on 11 Oct 2018

In Kubernetes there is a configmap used to give the main + parser configuration. If the change, the expectation is that fluent-bit will apply the changes.
Since there is no support for standard Unix 'HUP' behaviour of reloading config, the next best alternative is to exit and let the daemonset replication-controller restart. This should be seamless if the Tail plugin is used with DB & DB.Sync.
A new command-line flag (-C) and a new Service variable (Config_Watch: On|Off) are introduced to enable this behaviour.
This option may also be useful for systemd (which can restart on clean exit).

Open Support dynamic configuration #365 jolestar opened this issue on 24 Aug 2017 · 26 comments

oprotaso referenced this issue on 30 NOV 2018

Add sink resources #2 Merged

edsiper changed the title **Support-reload-configuration-files-Support dynamic configuration** on 11 Dec 2018

edsiper commented on 11 Dec 2018

I've changed the title of the issue since due to the feedback received internally and on GitHub what we need is dynamic configuration mechanisms which are more than a "config file reload".
After 1.0 we will focus on the design of this feature which involves having a parent process to monitor the engine plus other extensible interfaces to make this happens.

altenhof commented on 11 Dec 2018

Hi @edsiper,
thanks for the update. Just out of curiosity: What does "after 1.0" mean in terms of dates?

Support dynamic configuration #365

Open

jolestar opened this issue on 24 Aug 2017 · 39 comments



jolestar commented on 24 Aug 2017

Please support reload configuration files and do not need restart server.
I think there are two approach:

1. Catch SIGHUP signal
2. Offer a http api

50

Fluent Operator Features

- **Fluent Bit Management:** Deploy and destroy Fluent Bit DaemonSet automatically.
- **Fluentd Management:** Deploy and destroy Fluentd StatefulSet automatically.
- **Custom Configuration:** Select input/filter/output plugins via labels.
- **Dynamic Reloading:** Update configuration without rebooting Fluent Bit and Fluentd pods.
- **Multi-tenant log isolation:** Fluentd supports multi-tenant log isolation through label_router plugin.
- **Pluggable deployment components:** Either Fluent Bit or Fluentd can be deployed separately.



Fluent Operator facilitates the deployment of Fluent Bit & Fluentd and provides great flexibility in building a unified logging layer

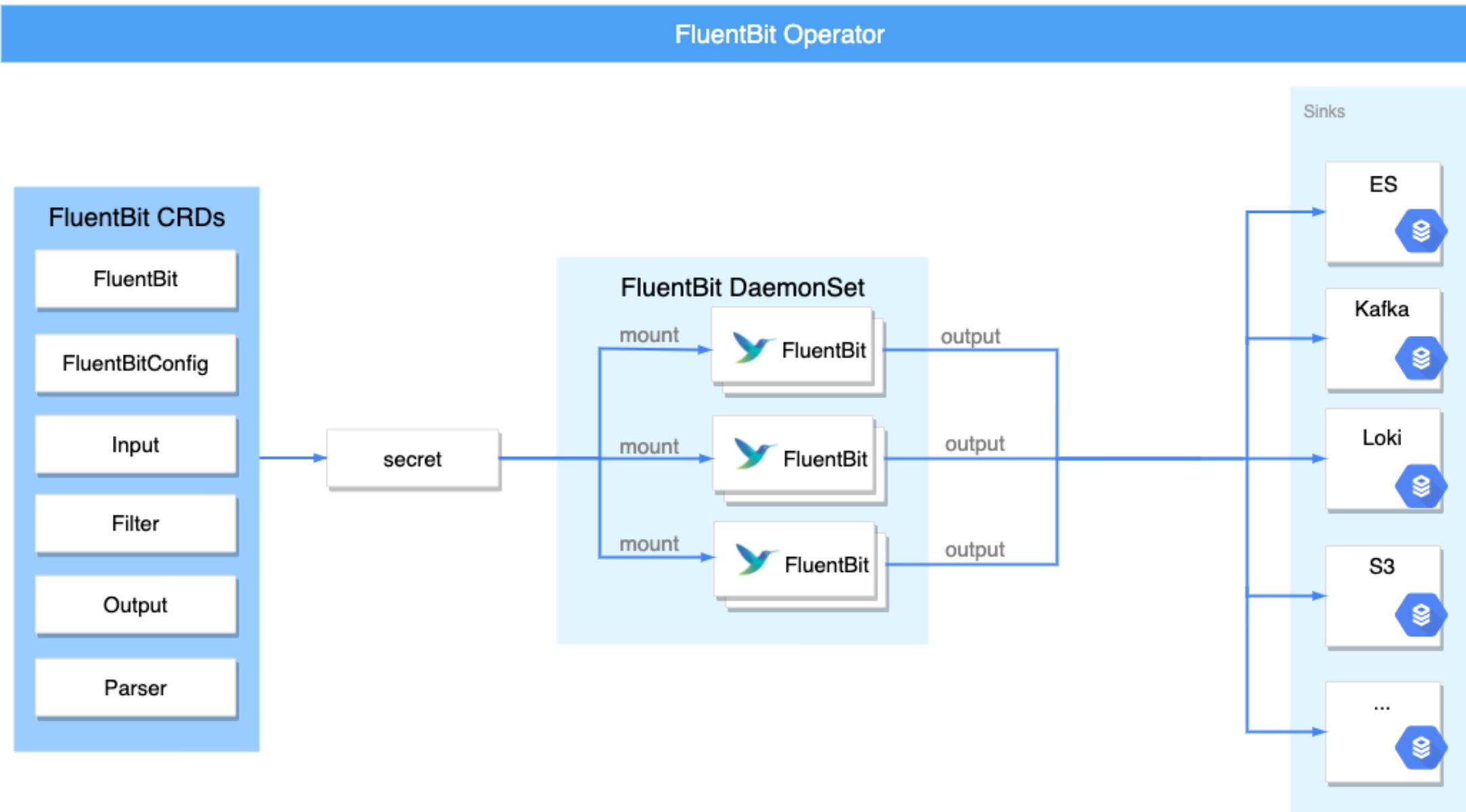
Fluent Operator Introduction - FluentBit Mgmt



KubeCon



CloudNativeCon





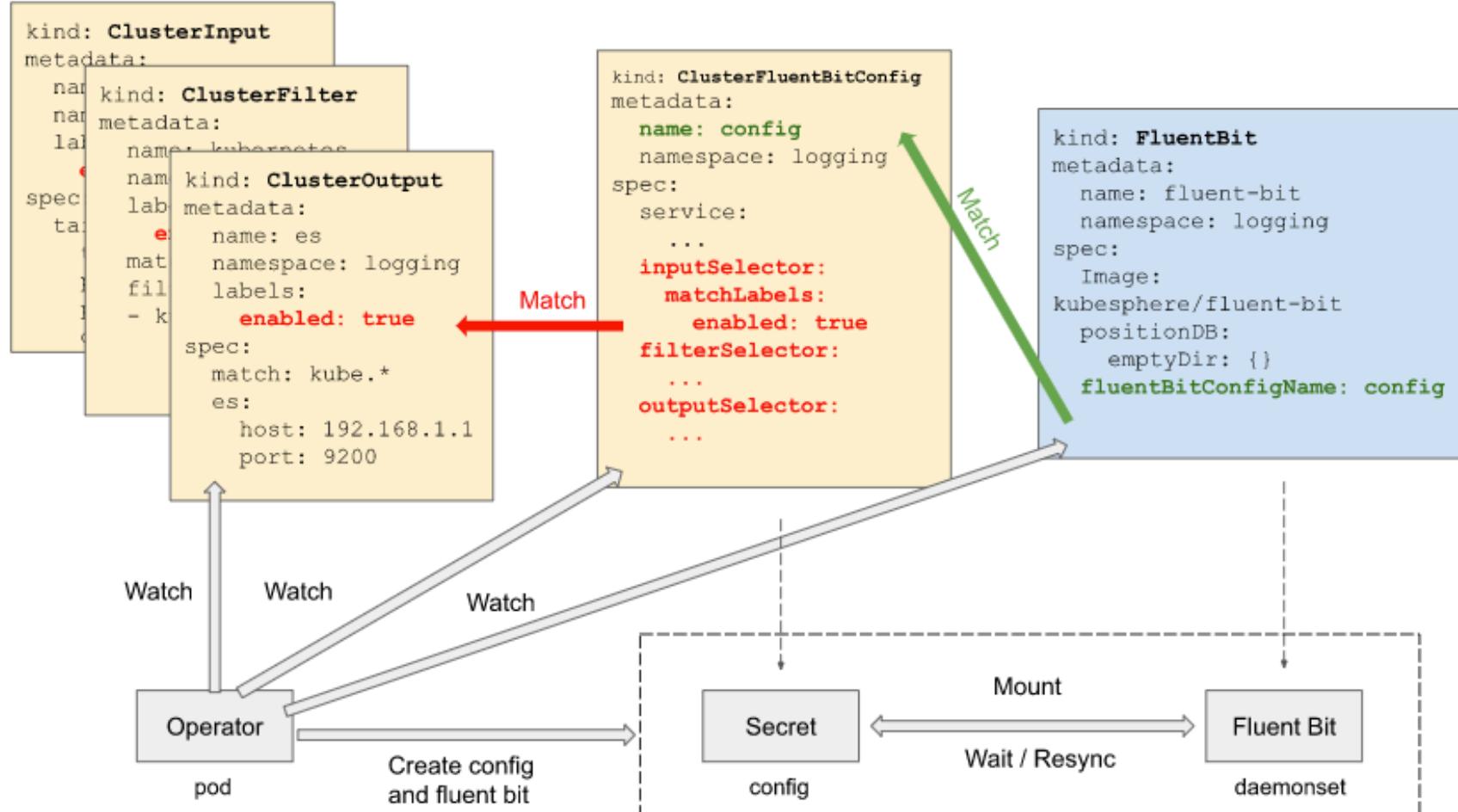
KubeCon



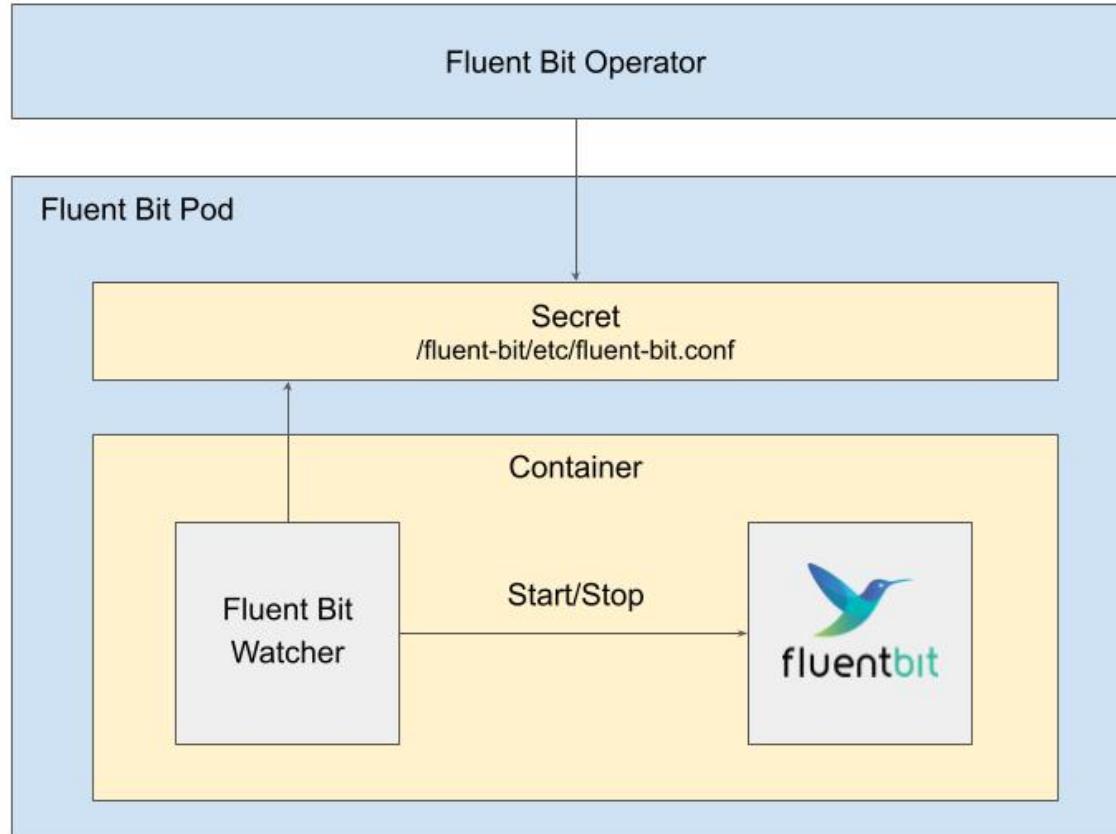
CloudNativeCon

Europe 2022

Fluent Operator Intro – FluentBit Mgmt



Fluent Operator Intro - FluentBit Operator



- Fluentd shouldn't be mandatory
- Fluent Bit dynamic config reloading is not yet supported:
 - <https://github.com/fluent/fluent-bit/issues/365>
 - <https://github.com/fluent/fluent-bit/pull/842>
- Use Fluent Bit Watcher to restart Fluent Bit process whenever Fluent Bit config changes without restarting the entire pod:
 - <https://github.com/fluent/fluent-operator/tree/master/cmd/fluent-watcher/fluentbit>

Fluent Operator General Architecture

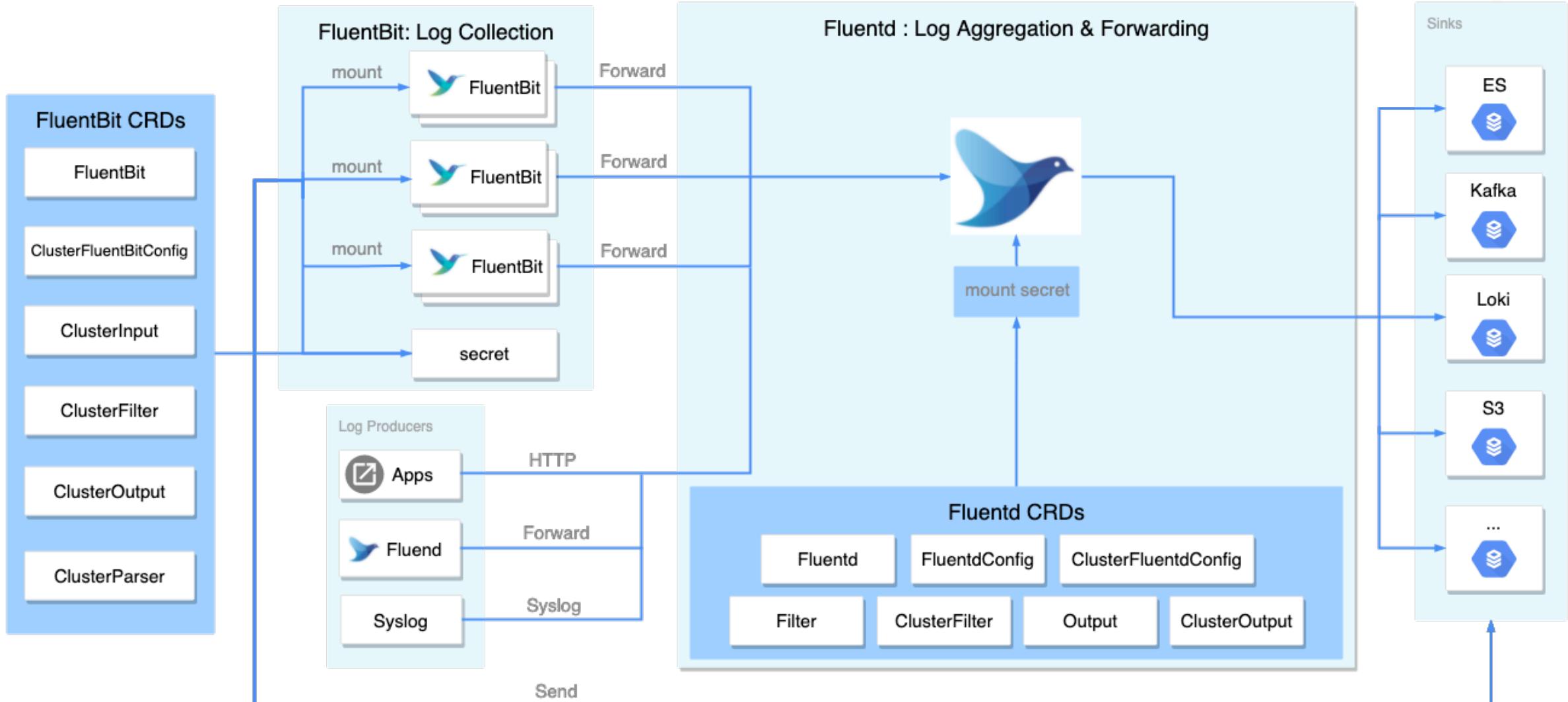


KubeCon

CloudNativeCon

2022

Fluent Operator : Build Cloud Native Log Processing Pipeline



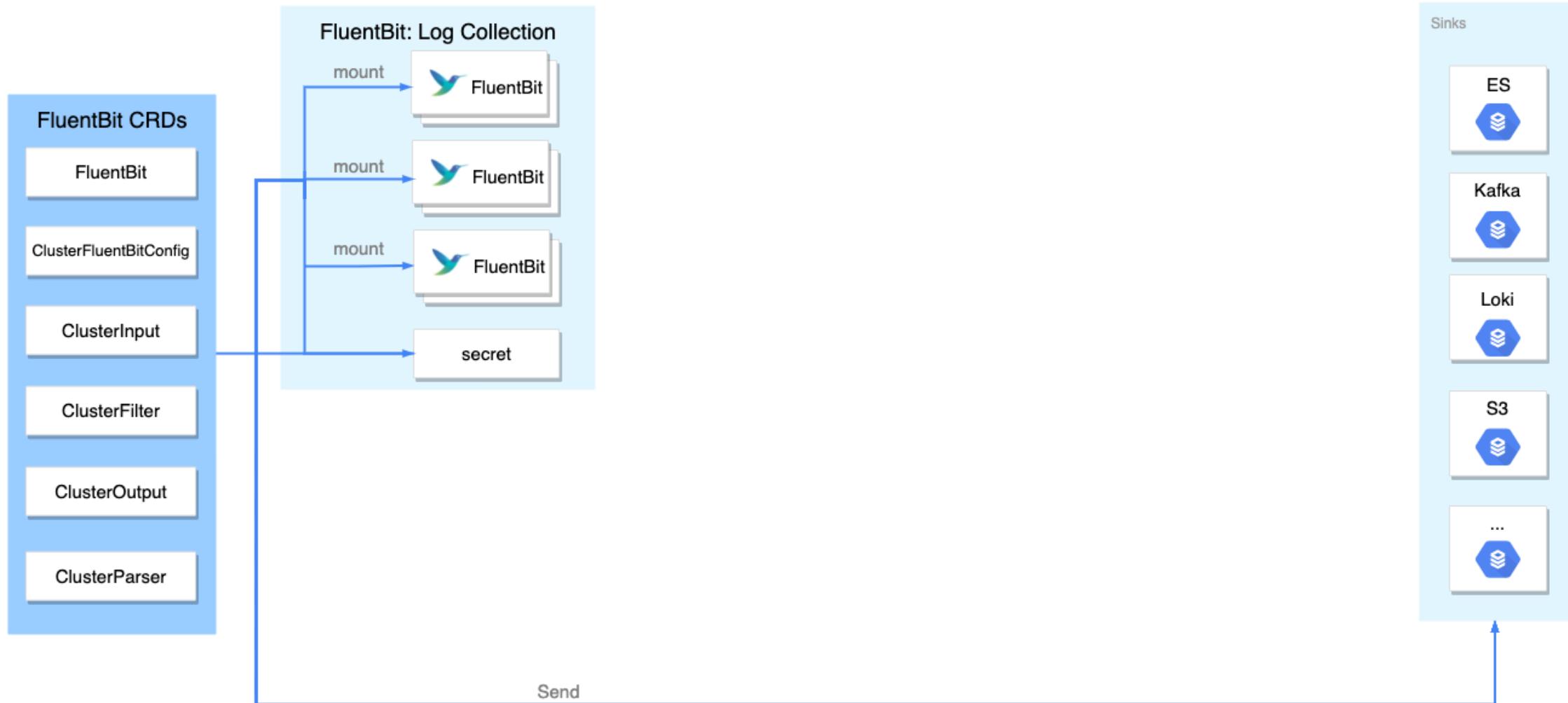
Use Fluent Operator as you want



- Fluent Bit Only mode
- Fluent Bit + Fluentd mode
- Fluentd only mode

Fluent Bit Only mode

Fluent Operator : Build Cloud Native Log Processing Pipeline





KubeCon



CloudNativeCon

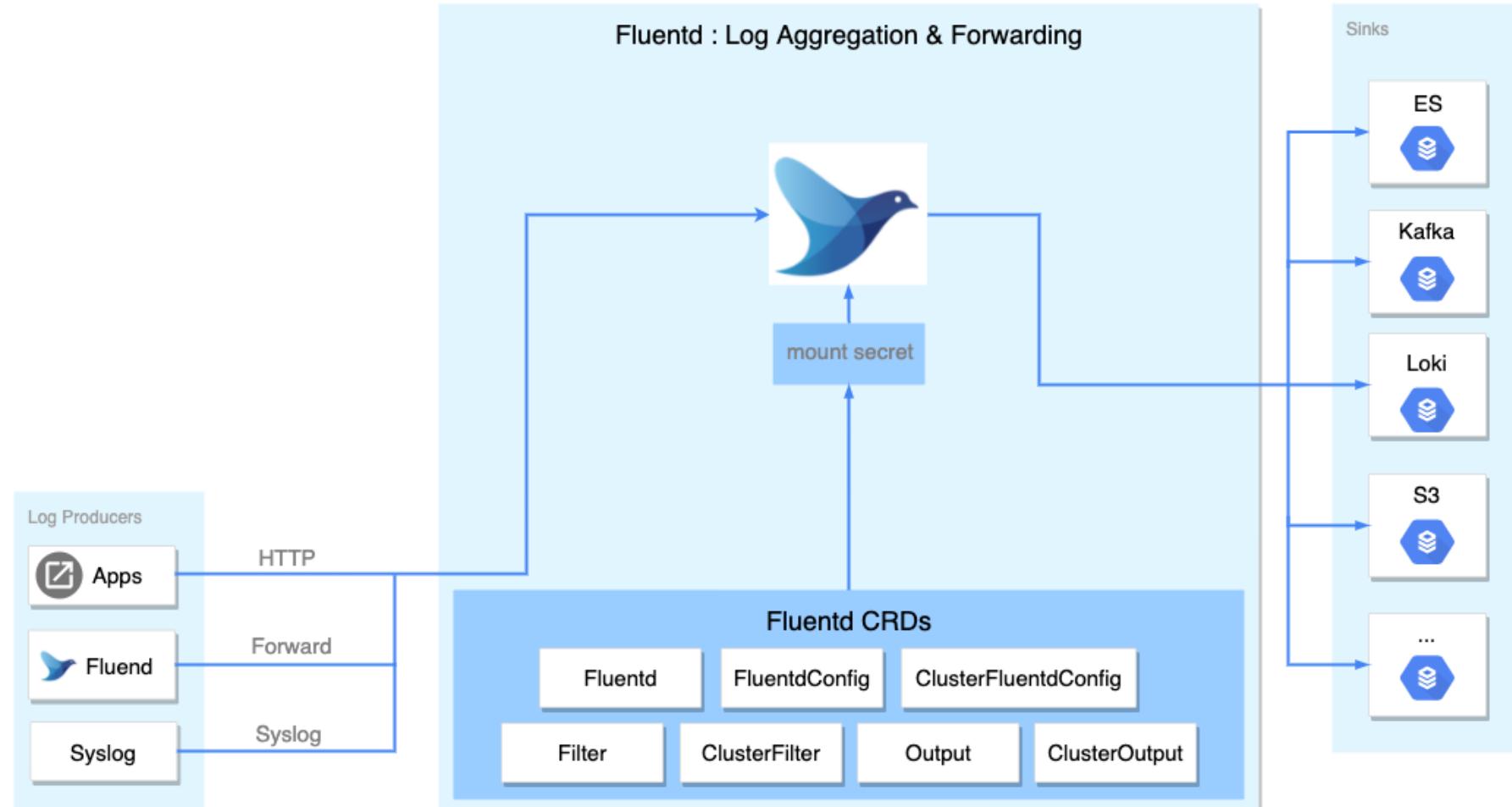
Europe 2022

```
cat <<EOF | kubectl apply -f -
apiVersion: fluentbit.fluent.io/v1alpha2
kind: ClusterFluentBitConfig
metadata:
  name: fluent-bit-config
  labels:
    app.kubernetes.io/name: fluent-bit
spec:
  service:
    parsersFile: parsers.conf
  inputSelector:
    matchLabels:
      fluentbit.fluent.io/enabled: "true"
      fluentbit.fluent.io	mode: "k8s"
  filterSelector:
    matchLabels:
      fluentbit.fluent.io/enabled: "true"
      fluentbit.fluent.io	mode: "k8s"
  outputSelector:
    matchLabels:
      fluentbit.fluent.io/enabled: "true"
      fluentbit.fluent.io	mode: "k8s"
_____
apiVersion: fluentbit.fluent.io/v1alpha2
kind: ClusterInput
metadata:
  name: tail
  labels:
    fluentbit.fluent.io/enabled: "true"
    fluentbit.fluent.io	mode: "k8s"
spec:
  tail:
    tag: kube.∗
    path: /var/log/containers/*.log
    parser: docker
    refreshIntervalSeconds: 10
    memBuflimit: 5MB
    skipLongLines: true
    db: /fluent-bit/tail/pos.db
    dbSync: Normal
```

```
apiVersion: fluentbit.fluent.io/v1alpha2
kind: ClusterFilter
metadata:
  name: kubernetes
  labels:
    fluentbit.fluent.io/enabled: "true"
    fluentbit.fluent.io	mode: "k8s"
spec:
  match: kube.∗
  filters:
    - kubernetes:
        kubeURL: https://kubernetes.default.svc:443
        kubeCAFile: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
        kubeTokenFile: /var/run/secrets/kubernetes.io/serviceaccount/token
        labels: false
        annotations: false
    - nest:
        operation: lift
        nestedUnder: kubernetes
        addPrefix: kubernetes_
    - modify:
        rules:
          - remove: stream
          - remove: kubernetes_pod_id
          - remove: kubernetes_host
          - remove: kubernetes_container_hash
    - nest:
        operation: nest
        wildcard:
          - kubernetes_∗
        nestUnder: kubernetes
        removePrefix: kubernetes_
_____
apiVersion: fluentbit.fluent.io/v1alpha2
kind: ClusterOutput
metadata:
  name: kafka
  labels:
    fluentbit.fluent.io/enabled: "false"
    fluentbit.fluent.io	mode: "k8s"
spec:
  matchRegex: (?:kube|service)\.(.*)
  kafka:
    brokers: my-cluster-kafka-bootstrap.kafka.svc:9091,my-cluster-kafka-
bootstrap.kafka.svc:9092,my-cluster-kafka-bootstrap.kafka.svc:9093
    topics: fluent-log
EOF
```

Fluentd only mode

Fluent Operator : Build Cloud Native Log Processing Pipeline





```
cat <<EOF | kubectl apply -f -
apiVersion: fluentd.fluent.io/v1alpha1
kind: Fluentd
metadata:
  name: fluentd-http
  namespace: fluent
  labels:
    app.kubernetes.io/name: fluentd
spec:
  globalInputs:
    - http:
        bind: 0.0.0.0
        port: 9880
  replicas: 1
  image: kubesphere/fluentd:v1.14.4
  fluentdCfgSelector:
    matchLabels:
      config.fluentd.fluent.io/enabled: "true"

—
apiVersion: fluentd.fluent.io/v1alpha1
kind: FluentdConfig
metadata:
  name: fluentd-only-config
  namespace: fluent
  labels:
    config.fluentd.fluent.io/enabled: "true"
spec:
  filterSelector:
    matchLabels:
      filter.fluentd.fluent.io	mode: "fluentd-only"
      filter.fluentd.fluent.io/enabled: "true"
  outputSelector:
    matchLabels:
      output.fluentd.fluent.io	mode: "true"
      output.fluentd.fluent.io/enabled: "true"
```

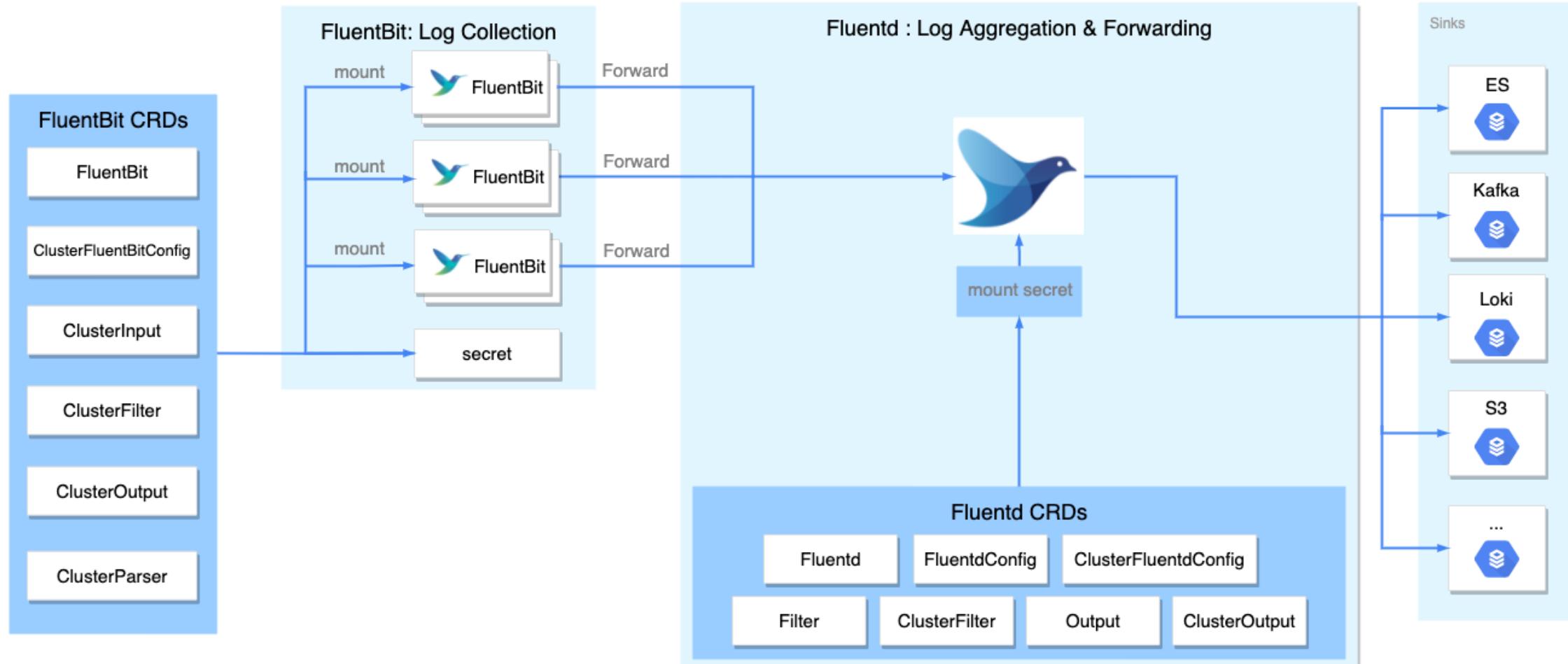
```
apiVersion: fluentd.fluent.io/v1alpha1
kind: Filter
metadata:
  name: fluentd-only-filter
  namespace: fluent
  labels:
    filter.fluentd.fluent.io	mode: "fluentd-only"
    filter.fluentd.fluent.io/enabled: "true"
spec:
  filters:
    - stdout: {}

—
apiVersion: fluentd.fluent.io/v1alpha1
kind: Output
metadata:
  name: fluentd-only-stdout
  namespace: fluent
  labels:
    output.fluentd.fluent.io	mode: "true"
    output.fluentd.fluent.io/enabled: "true"
spec:
  outputs:
    - stdout: {}

EOF
```

Fluent Bit + Fluentd mode

Fluent Operator : Build Cloud Native Log Processing Pipeline



Demo: Using Fluent Operator to process logs

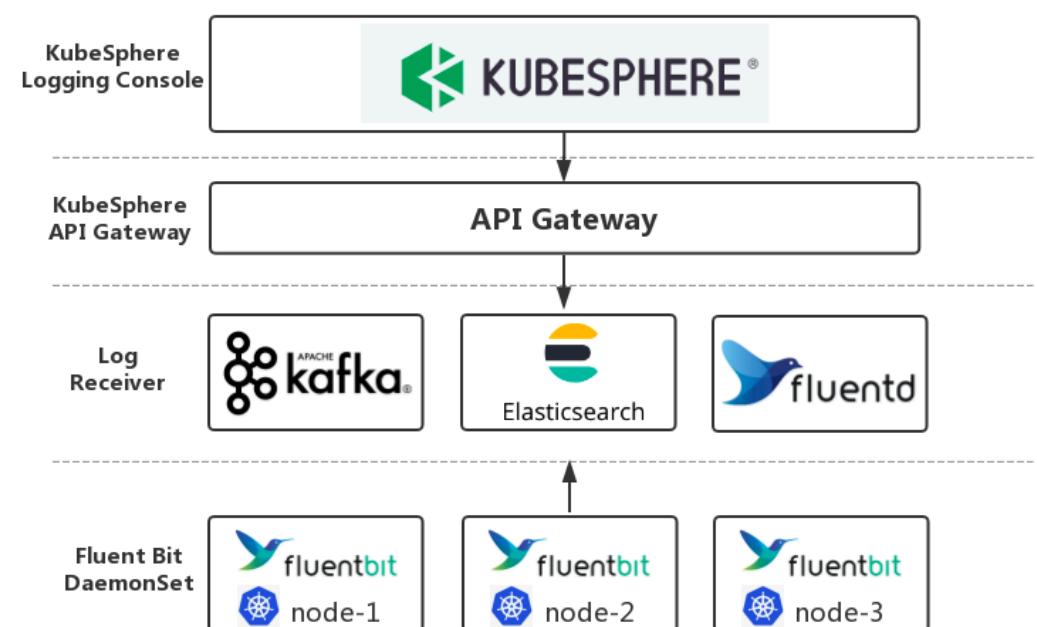
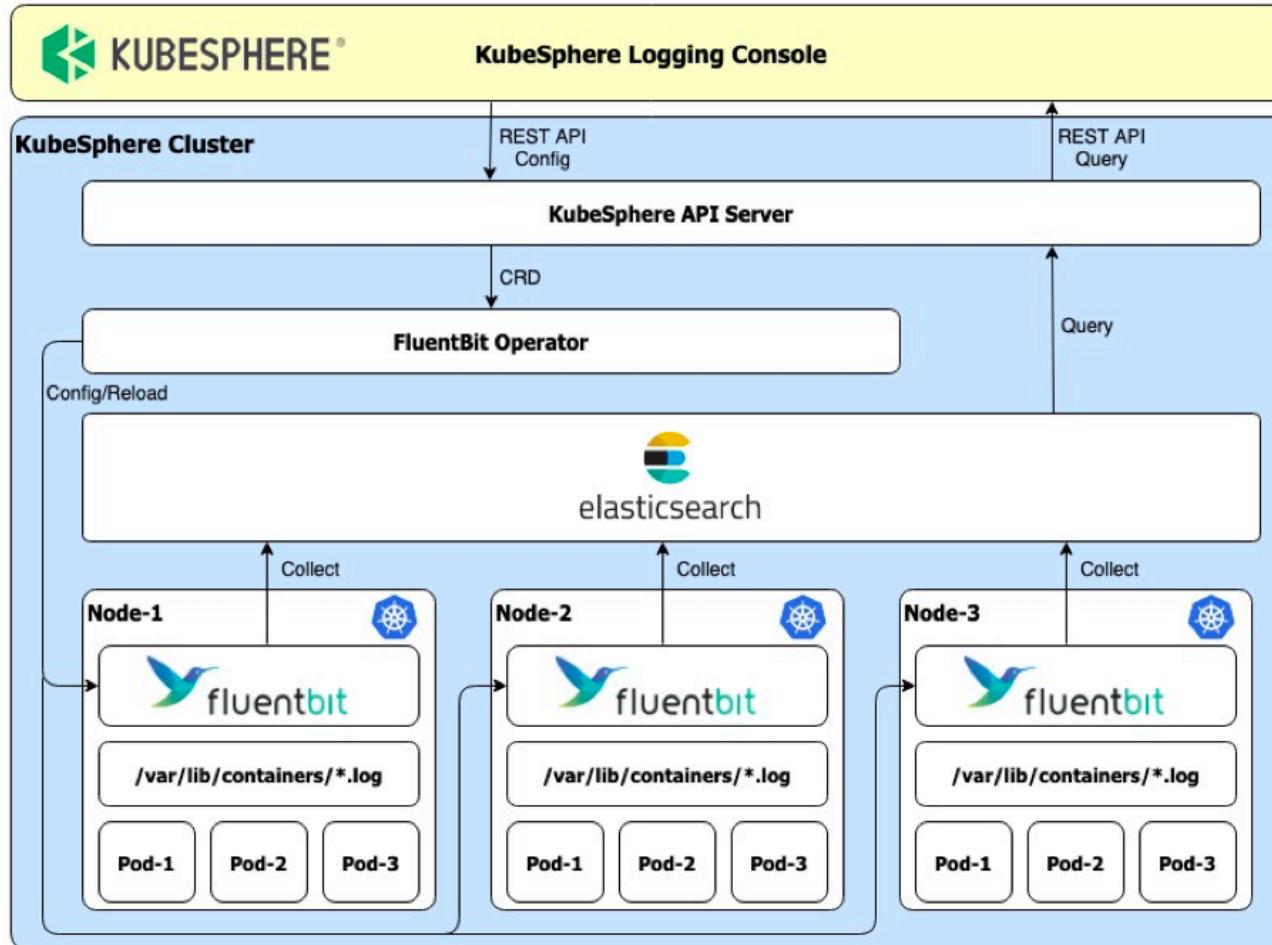


fluent-operator-walkthrough

- fluent-operator-walkthrough
 - Prerequisite
 - Install Fluent Operator
 - Deploy Fluent Bit and Fluentd
 - Deploy Fluent Bit
 - Deploy Fluentd
 - Fluent Bit Only mode
 - Using Fluent Bit to collect kubelet logs and output to Elasticsearch
 - Using Fluent Bit to collect K8s application logs and output to Kafka
 - Fluent Bit + Fluentd mode
 - Forward logs from Fluent Bit to Fluentd
 - Enable Fluentd Forward Input plugin to receive logs from Fluent Bit
 - ClusterFluentdConfig: Fluentd cluster-wide configuration
 - FluentdConfig: Fluentd namespaced-wide configuration
 - Use cluster-wide and namespaced-wide FluentdConfig together
 - Use cluster-wide and namespaced-wide FluentdConfig together in multi-tenant scenarios
 - Route logs to different Kafka topics based on namespace
 - Using buffer for Fluentd output
 - Fluentd only mode
 - Use fluentd to receive logs from HTTP and output to stdout
 - How to check the configuration and data

PromCon
North America 2022

Case Study: Fluent Operator in KubeSphere



Follow us!



github.com/kubesphere

PromCon
North America 2022



twitter.com/kubesphere



kubesphere.io



kubesphere.slack.com