# 로그수집, 코드로 관리하고 쉽게 적용하자.

임성일

- Software Developer @ SK Telecom
- TACO development
  - Logging / Monitoring / Alerting
  - GitOps tools
  - Micro serivces
- OSS Contributor (~2019)
  - Openstack
  - Fluentbit
- Devocean (SK Tech 전문가그룹)
- E-Mail: si.im@sk.com, usnexp@gmail.com

Developers' Ocean 개발자들을 위한 영감의 바다

Devotion 헌신,몰두,전념

Technology for Everyone 모두를 위한 기술

# Contents

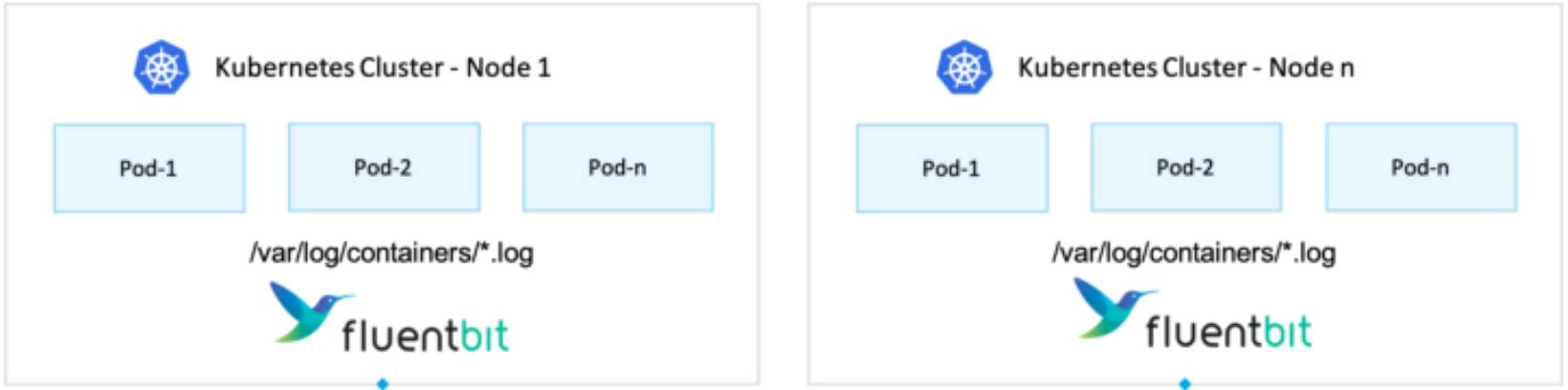
**Central Logging in Kubernetes**

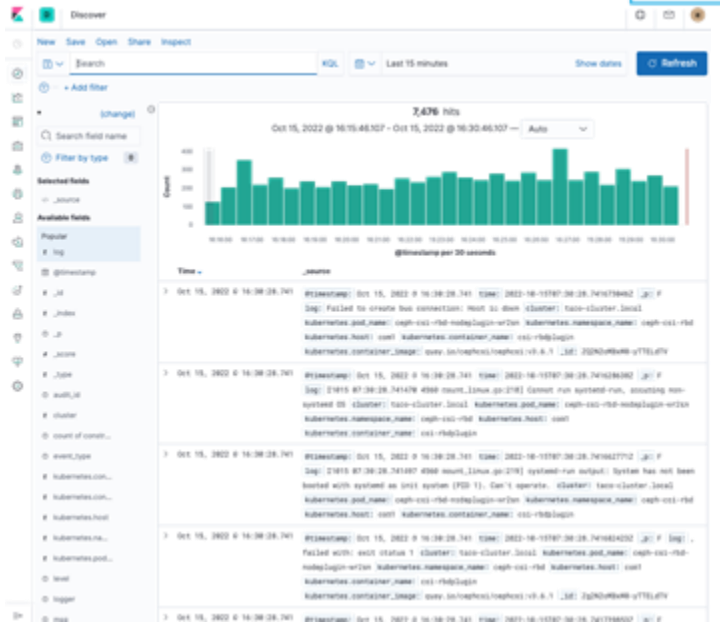
**Fluent bit & Operator**

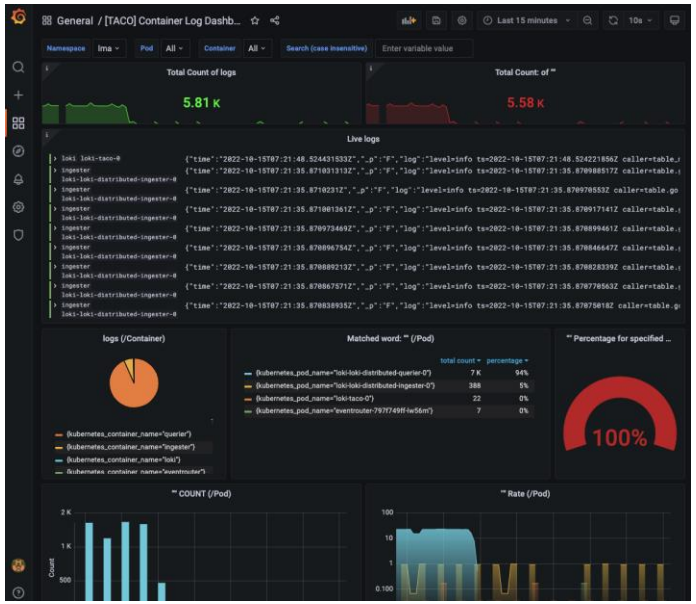
**fluentbit-resource chart**

**적용사례** (TACO) **및** Demo.

# Central logging in Kubernetes (TACO LMA, EFK)





# Fluent/Fluent-bit ?

- CNCF graduate project
- Log collector
- Plug-in structure
- fluentd, fluent-bit

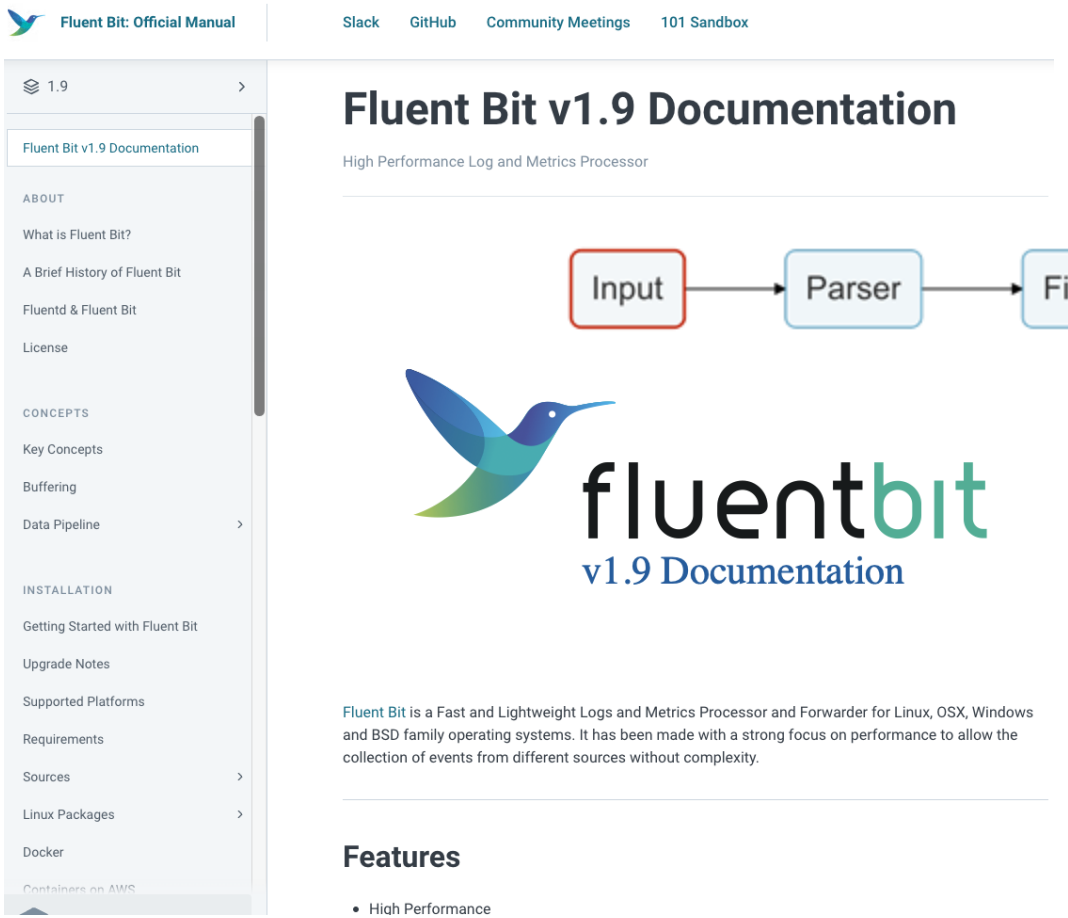
Fluent Bit is an open source and multi-platform log processor tool which aims to be a generic Swiss knife for logs processing and distribution.

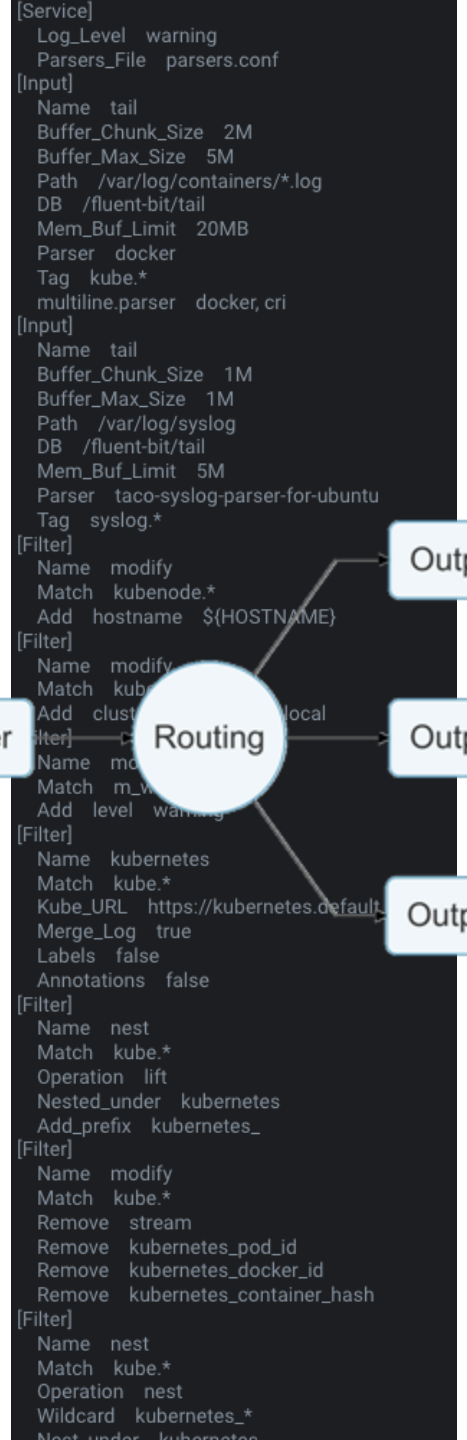
Nowadays the number of sources of information in our environments is ever increasing. Handling data collection at scale is complex, and collecting and aggregating diverse data requires a specialized tool that can deal with:

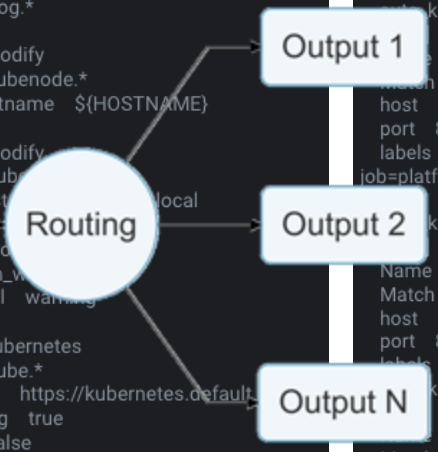
- Different sources of information
- Different data formats
- Data Reliability
- Security
- Flexible Routing
- Multiple destinations

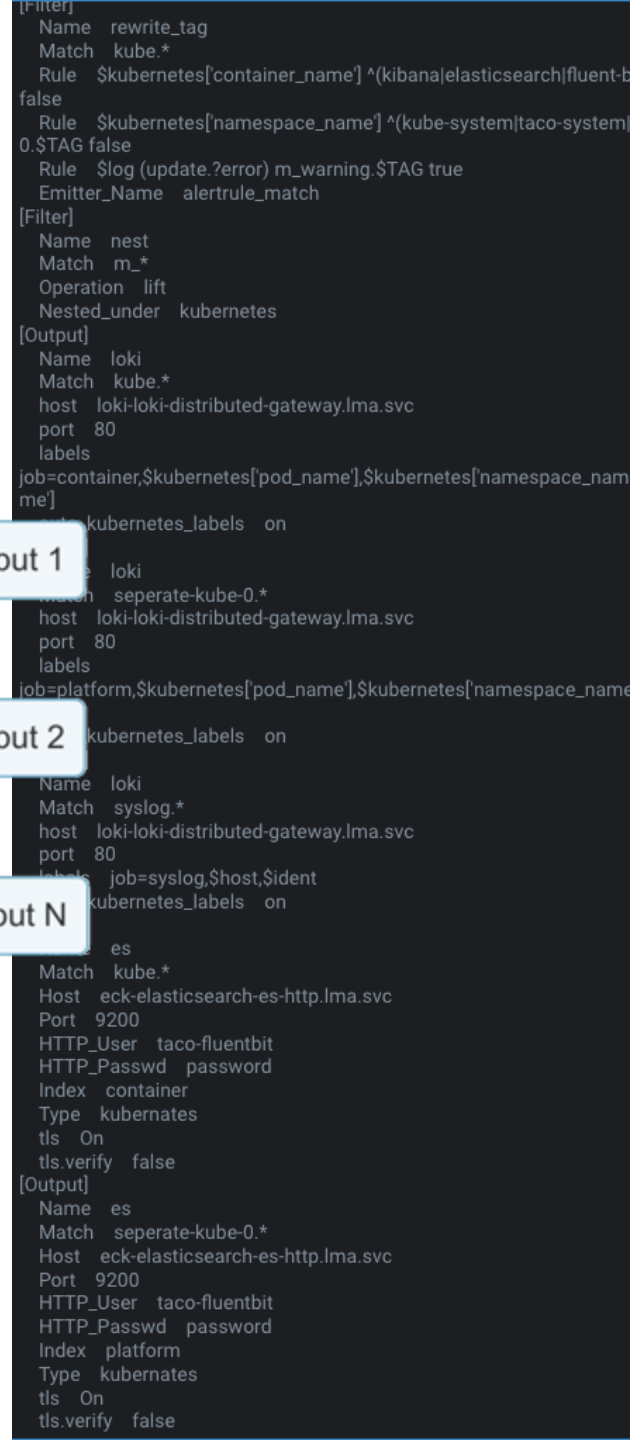
Fluent Bit has been designed with performance and low resources consumption in mind.

# Fluent-bit 로그수집 설정



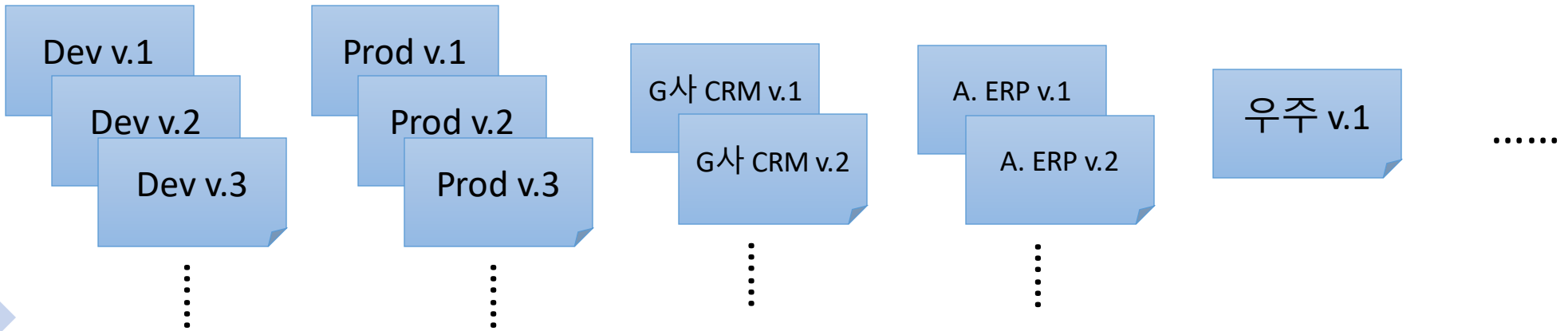


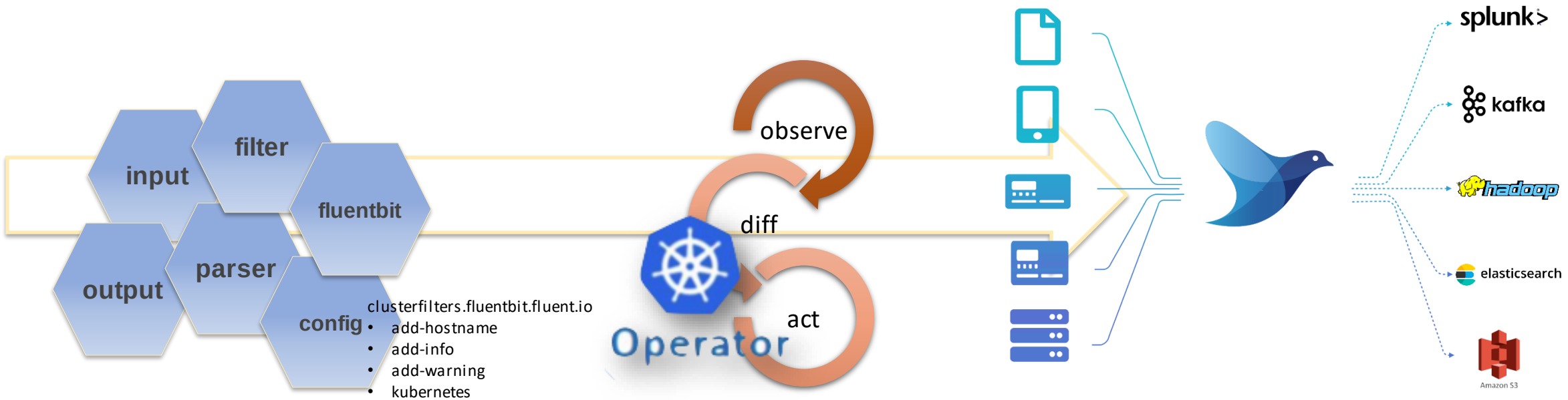



# 변경 및 형상관리의 어려움

- DevOps가 되면서 어플리케이션과 인프라의 변경이 잦아졌고
  - Kubernetes환경은  이를 지원하기 위한 최적의 도구로 잦은 변경이 발생하고 이에 따라 로그수집의 기준을 변경할 필요가 발생
- 솔루션으로 TACO는 다양한 환경으로 설치되고 이에 대한 관리 지원필요



# Fluentbit operator




clusterfilters.fluentbit.fluent.io
- add-hostname
- add-info
- add-warning
- kubernetes
- simplify

clusterfluentbitconfigs.fluentbit.fluent.io
- fluent-bit-config

clusterinputs.fluentbit.fluent.io
- input-kube
- input-syslog

clusteroutputs.fluentbit.fluent.io
- fbcr-taco-loki-taco-loki-kube
- fbcr-taco-loki-taco-loki-seperate-kube-0
- fbcr-taco-loki-taco-loki-syslog
- fbcr-taco-taco-es-container
- fbcr-taco-taco-es-platform

clusterparsers.fluentbit.fluent.io
- taco-syslog-parser-for-ubuntu

fluentbits.fluentbit.fluent.io
- fluent-bit

```
siim@dev:~$ kubectl get po -n lma | grep fluent
fluent-bit-8f75b                                  1/1     Running     0
fluent-bit-9mjw2                                  1/1     Running     0
fluent-bit-ltz8x                                  1/1     Running     0
fluent-bit-lwwxd                                  1/1     Running     0
fluent-bit-msn6w                                  1/1     Running     0
fluent-bit-s1p59                                  1/1     Running     0
fluent-bit-whbds                                  1/1     Running     0
```

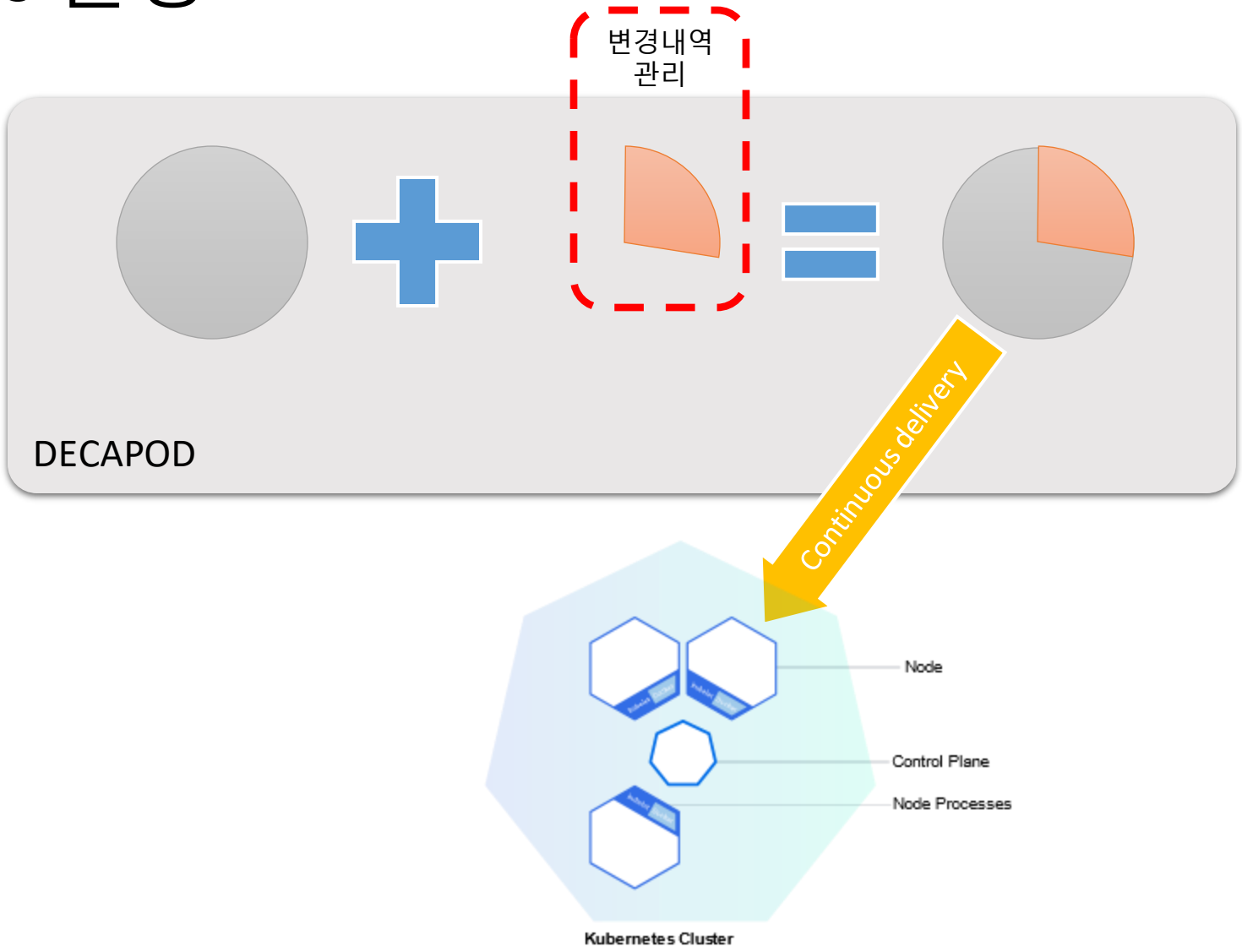
# Decapod 를 통한 Gitops 활용

Decapod는

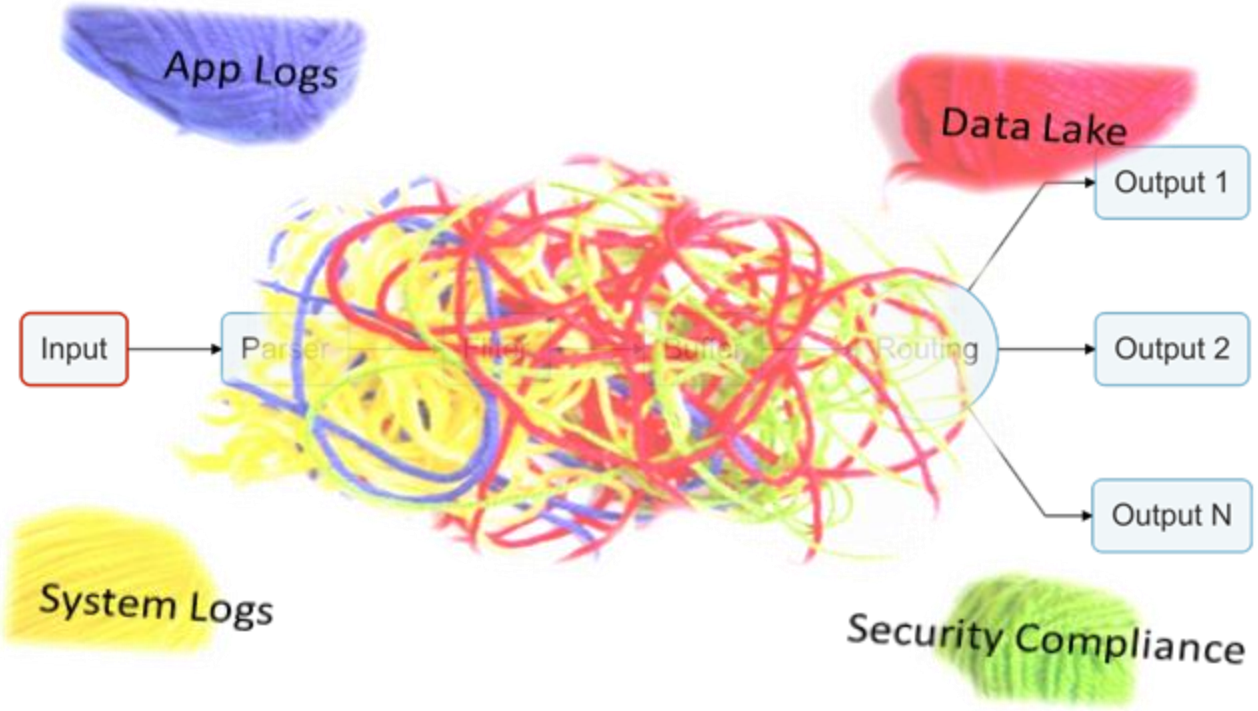
- 기본 값을 갖고 있으며 필요한 부분만 재지정하여 전체를 구성
- CD기능을 통해 지속적인 배포
- Git을 통한 버전관리

이를 통해 얻는 효과

- Site별 최소 변경기능만 정의 및 관리
- 설정에 대한 버전관리
- 변경내역의 자동반영



# 설정의 어려움



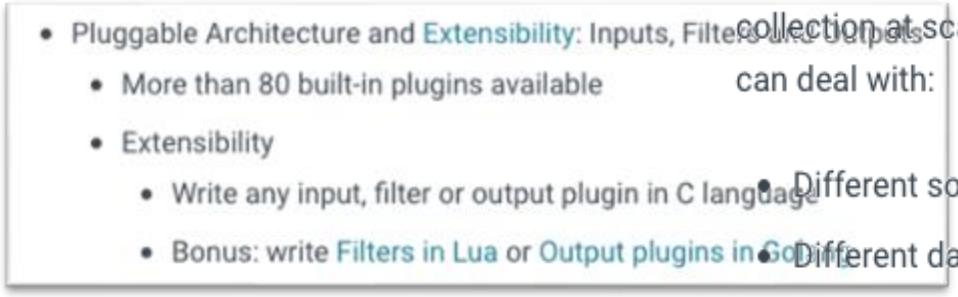

Log의 종류가 많아지고 다양한 처리가 추가되면 복잡도가 지속적으로 증가함

```
[Service]
  Log_Level    warning
  Parsers_File    parsers.conf
[Input]
  Name    tail
  Buffer_Chunk_Size    2M
  Buffer_Max_Size    5M
  Path    /var/log/containers/*.log
  DB    /fluent-bit/tail
  Mem_Buf_Limit    20MB
  Parser    docker
  Tag    kube.*
  multiline.parser    docker, cri
[Input]
  Name    tail
  Buffer_Chunk_Size    1M
  Buffer_Max_Size    1M
  Path    /var/log/syslog
  DB    /fluent-bit/tail
  Mem_Buf_Limit    5M
  Parser    taco-syslog-parser-for-ubuntu
  Tag    syslog.*
[Filter]
  Name    modify
  Match    kubenode.*
  Add    hostname    ${HOSTNAME}
[Filter]
  Name    modify
  Match    kube*
  Add    cluster    taco-cluster.local
[Filter]
  Name    modify
  Match    m_warning.*
  Add    level    warning
[Filter]
  Name    kubernetes
  Match    kube.*
  Kube_URL    https://kubernetes.default.svc:443
  Merge_Log    true
  Labels    false
  Annotations    false
[Filter]
  Name    nest
  Match    kube.*
  Operation    lift
  Nested_under    kubernetes
  Add_prefix    kubernetes_
[Filter]
  Name    modify
  Match    kube.*
  Remove    stream
  Remove    kubernetes_pod_id
  Remove    kubernetes_docker_id
  Remove    kubernetes_container_hash
[Filter]
  Name    nest
  Match    kube.*
  Operation    nest
  Wildcard    kubernetes_*
  Nest_under    kubernetes
```

```
[Filter]
  Name    rewrite_tag
  Match    kube.*
  Rule    $kubernetes['container_name'] ^(kibana|elasticsearch|fluent-b
false
  Rule    $kubernetes['namespace_name'] ^(kube-system|taco-system|
0.$TAG false
  Rule    $log (update.?error) m_warning.$TAG true
  Emitter_Name    alertrule_match
[Filter]
  Name    nest
  Match    m_*
  Operation    lift
  Nested_under    kubernetes
[Output]
  Name    loki
  Match    kube.*
  host    loki-loki-distributed-gateway.lma.svc
  port    80
  labels
job=container,$kubernetes['pod_name'],$kubernetes['namespace_nam
me']
  auto_kubernetes_labels    on
[Output]
  Name    loki
  Match    seperate-kube-0.*
  host    loki-loki-distributed-gateway.lma.svc
  port    80
  labels
job=platform,$kubernetes['pod_name'],$kubernetes['namespace_name
me']
  auto_kubernetes_labels    on
[Output]
  Name    loki
  Match    syslog.*
  host    loki-loki-distributed-gateway.lma.svc
  port    80
  labels    job=syslog,$host,$ident
  auto_kubernetes_labels    on
[Output]
  Name    es
  Match    kube.*
  Host    eck-elasticsearch-es-http.lma.svc
  Port    9200
  HTTP_User    taco-fluentbit
  HTTP_Passwd    password
  Index    container
  Type    kubernates
  tls    On
  tls.verify    false
[Output]
  Name    es
  Match    seperate-kube-0.*
  Host    eck-elasticsearch-es-http.lma.svc
  Port    9200
  HTTP_User    taco-fluentbit
  HTTP_Passwd    password
  Index    platform
  Type    kubernates
  tls    On
  tls.verify    false
```

# 무한의 자유도가 필요한가?

- 복잡한 이유
  - 다양한 전처리 후처리가 가능하도록 만들어짐




- 그런데 우리는? 제한된 환경에서 사용
  - Kubernetes
  - Application, system log…
  - Tail input plugin
  - Elasticsearch, kafka, loki…

Fluent Bit is an open source and multi-platform log processor tool which aims to be a generic Swiss knife for logs processing and distribution.

Nowadays the number of sources of information in our environments is ever increasing. Handling data collection at scale is complex, and collecting and aggregating diverse data requires a specialized tool that can deal with:

- Different sources of information
- Different data formats
- Data Reliability
- Security
- Flexible Routing
- Multiple destinations

Fluent Bit has been designed with performance and low resources consumption in mind.

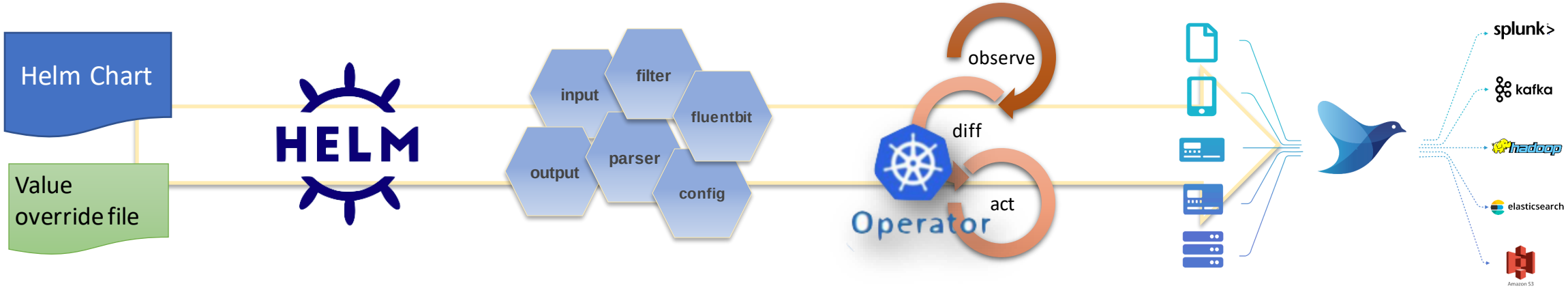
- 필요한 부분만으로 자유도를 제약한 간결한 구조의 지원이 필요

# fluentbit-resource Helm chart

- Fluentbit-operator에 사용자자원을 배포
  - Input
  - Output
  - Parser
  - Filter
  - fluentbit
- 수집 대상: log
  - Kubernetes 위에서 실행되는 워크로드에 대한 로그
  - Kubernetes 를 운영하기위한 기반 infra에서 발생하는 로그

[ 주요 특징 ]

- 원 로그의 지정과 이에 대한 처리, 그리고 저장위치를 한번에 지정
- 하나의 소스에서 발생한 로그를 조건에 따라 다양한 저장위치로 저장
- 저장소는 ElasticSearch를 기본으로하고 kafka, loki 등을 지원
- prometheus와 연동하여 로그 기반으로 알람생성
- ElasticSearch에 초기화
  - 접근용 전용계정생성
  - index 관리 기능 설정
  - index의 template 지정



# 저장대상 지정

- outputs
- 최종적으로 로그를 전달할 output에 대한 정보와 초기화 기능 설정
- 지원대상
  - Elasticsearch (outputs.es)
  - Loki (outputs.loki)
  - Kafka (outputs.kafka)
- Elasticsearch 초기화 기능
  - 로그수집 전용 사용자 생성
  - schema 지정 (template)
  - ilm(Index Lifecycle Management) 지정

Grafana loki

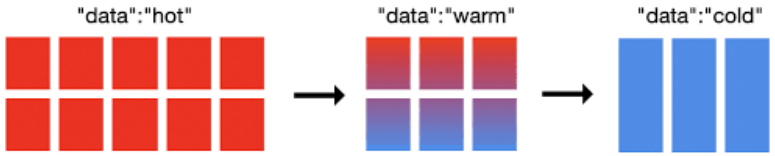
```
28  outputs:
29    loki:
30    - name: taco-loki
31      host: loki
32      port: 3100
```

kafka

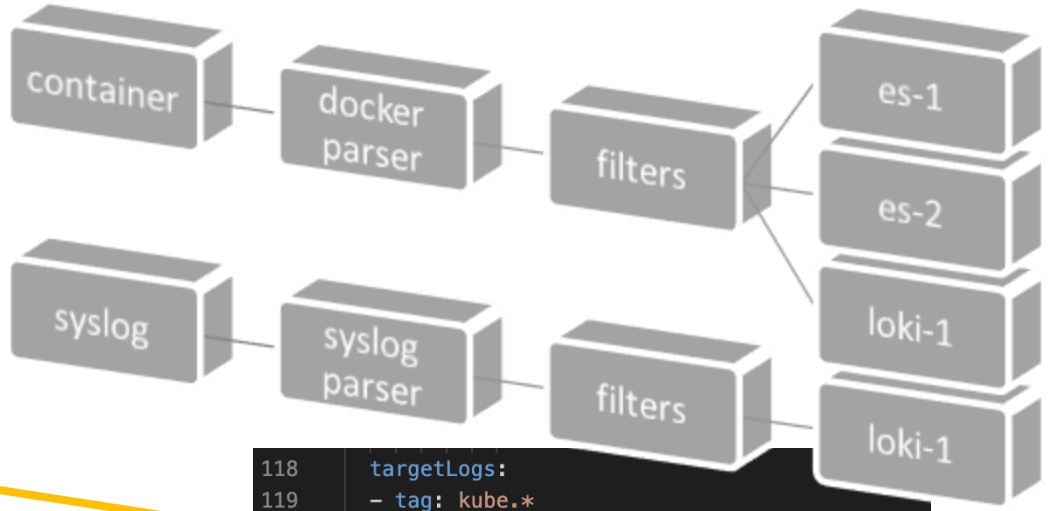
```
28  outputs:
29    es:
30    - name: taco-es
31      host: eck-elasticsearch-es-http
32      port: 9200
33
34      dedicatedUser:
35        username: taco-fluentbit
36        password: password
37        elasticPasswordSecret: eck-elasticsearch-es-elastic-user
38
39      template:
40        enabled: true
41        ilms:
42        - name: hot-delete-14days
43          json:
44            policy:
45              phases:
46                hot:
47                  actions:
48                    rollover:
49                      max_size: 30gb
50                      max_age: 1d
51                      max_docs: 5000000000
52                    set_priority:
53                      priority: 100
54                delete:
55                  min_age: 14d
56                  actions:
57                    delete: {}
```

elasticsearch

```
90    templates:
91    - name: platform
92      json:
93        index_patterns: "platform*"
94        settings:
95          refresh_interval: 30s
96          number_of_shards: 3
97          number_of_replicas: 1
98          index.lifecycle.name: hot-delete-14days
99          index.lifecycle.rollover_alias: platform
```



# 로그별 흐름지정




- targetLogs:
- 로그별 수집위치부터 저장위치까지 기록
- 설정필드
  - 이름지정(name)
  - 대상로그(path): tail
  - 전처리 파서(parser): docker나 syslog, 또는 parser추가를 통한 지정
  - 태그정의(tag)
  - 버퍼관련 지정(bufferChunkSize, bufferMaxSize,memBufLimit)
  - 로그를 저장할 것인지 지정(do_not_store_as_default): 다중저장의 경우 사용됨
- 단일 저장소 (1 source -> 1 store)
  - es_name, loki_name: 타입별 이름을 지정
  - index: Index(es) 또는 stream(loki) 지정
- 다중 저장소 (1 source -> n store)
  - 리스트 형태로 조건을 정의하고 보낼 저장소를 지정
  - 조건은 필드별로 regex를 통해 설정

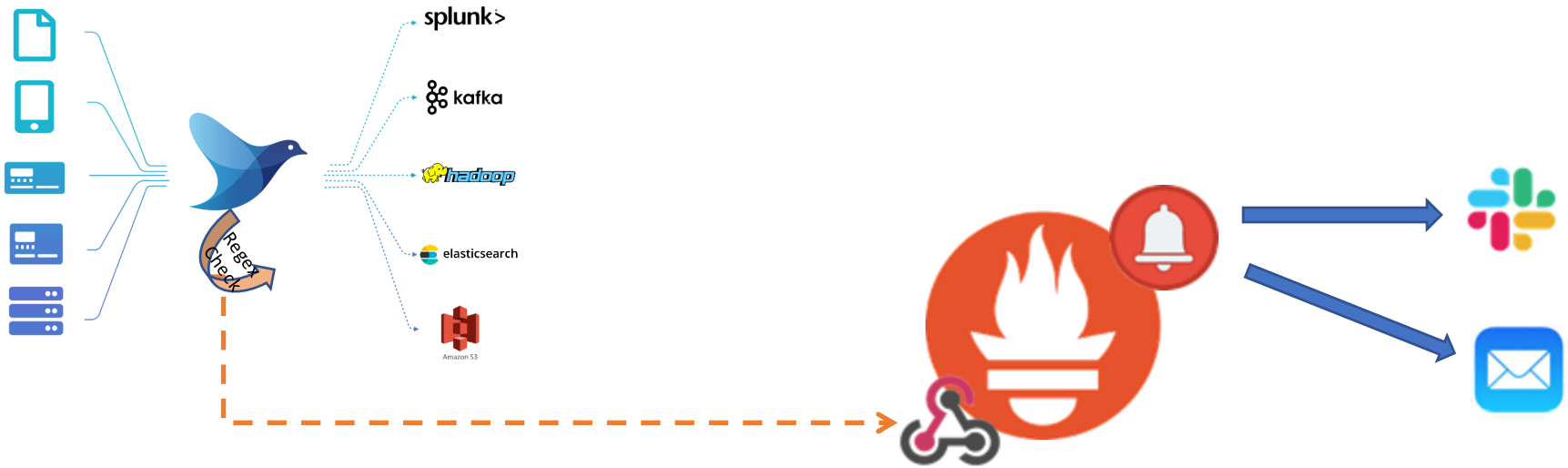
```
118   targetLogs:
119   - tag: kube.*
120     bufferChunkSize: 2M
121     bufferMaxSize: 5M
122     do_not_store_as_default: false
123     es_name: taco-es
124     index: container
125     memBufLimit: 20MB
126     multi_index:
127     - index: platform
128       es_name: taco-es
129       key: $kubernetes['namespace_name']
130       value: kube-system|taco-system|lma|argo
131     parser: docker
132     path: /var/log/containers/*.log
133     type: kubernates
134     extraArgs:
135       multilineParser: docker, cri
136   - tag: syslog.*
137     es_name: taco-es
138     index: syslog
139     parser: taco-syslog-parser-for-ubuntu
140     path: /var/log/syslog
141     type: syslog
```

# 로그기반 알람설정

```
143   alerts:
144     enabled: true
145     namespace: taco-system
146     message: |-
147       {{ $labels.container }} in {{ $labels.pod }} ({{ $labels.taco_cluster }}/{{ $
148     summary: |-
149       {{ $labels.container }} in {{ $labels.pod }} ({{ $labels.taco_cluster }}/{{ $
150     rules:
151     # - name: critical-example
152     #   severity: critical
153     #   regex: "OOM Killed|Evict"
154     - name: warning-example
155       severity: warning
156       regex: "update.?error"
```

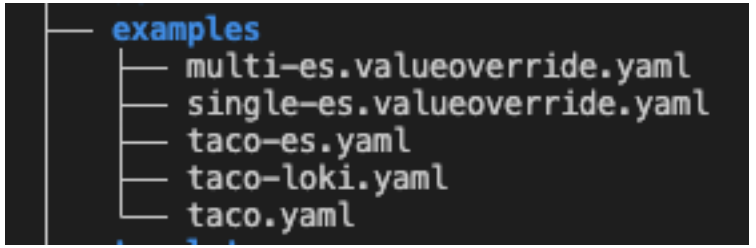
- TACO LMA를 설치한 경우 로그기반 알람 가능
- 문구의 점검을 위한 범위 지정: namespace
- 알람에 대한 문구를 정의하여 알람 발생시 메시지를 정의
  - alerts.message
  - Alerts.summary
- 알람룰에 대한 정의
  - 로그의 중요도 설정: prometheus의 기준에 따르며 page / critical / warning / info
  - regex: 정규식을 지정하면 수집되는 모든 로그를 검사하여 일치하면 알람발생
- Exporter 활성화 필요

```
79   logExporter:
80     enabled: true
81     serviceMonitor:
82       enabled: true
83     spec:
84       nodeSelector:
85         taco-lma: enabled
```

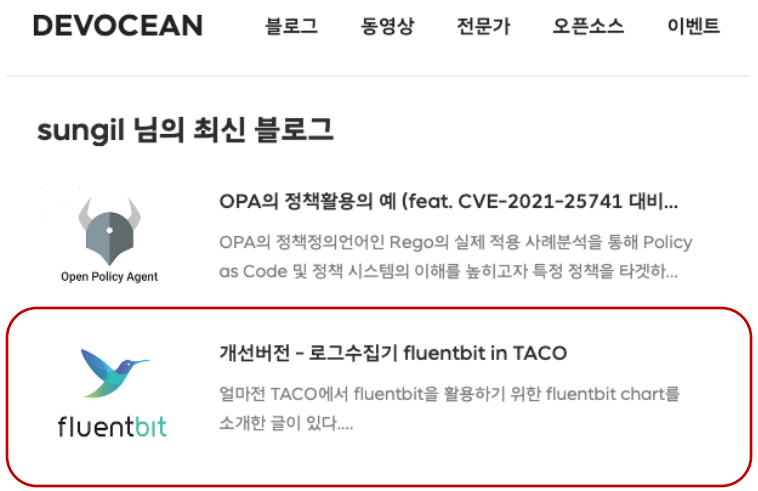


# 사용자를 위한 지원

- 해당 Helm chart에는 사용자들이 쉽게 접근할 수 있도록 value override 예제들이 포함됨
  - 단일 es를 사용하는 예제 (single-es)
  - 다중 es를 사용하는 예제 (multi-es)
  - Loki를 사용하는 예제 (loki)




- Devocean 블로그
  - 개선버전 - 로그수집기 fluentbit in TACO
  - https://devocean.sk.com/blog/techBoardDetail.do?ID=164148



# 사용해보기 (Demo)

- kubernetes에서 동작하는 모든 컨테이너 로그와 시스템로그 수집
- 저장소는 elasticsearch, loki가 존재 (설치포함)
- 모든 container 로그는 elasticsearch의 container 인덱스에 저장
- 네임스페이스 kube-system에서 발생하는 로그는 elasticsearch의 system 인덱스에 저장

# Summary

- fluent bit은 log 수집을 위한 가볍고 많은 기능을 제공하는 툴
- 설정의 어려움과 다양한 환경에 적용하기 위한 도전이 있음
- fluentbit-resource helm chart는 이러한 어려움을 해소하고자 만들어진 유틸리티
- kubernetes에서 발생하는 로그들을 Es, loki 로 일관적으로 흐르도록 할 수 있음
- Fluent bit을 사용하는 것으로 다양한 확장이 가능하고 openinfradev (일명 TACO) 프로젝트를 통해 기여해주세요

Thanks