Effective Logging in Multi-Tenant Kubernetes Environment

Benjamin Huo
Lead of KubeSphere Observability team

Dan Ma
Senior SE of KubeSphere Observability team
How to debug your K8S workloads?

- `kubectl logs pod1 -c container2`
- Cloud providers’ logging solutions: stackdriver, cloudwatch
- ISVs’ logging solution: Splunk, Sumo Logic, Datadog
- OSS logging solutions: EFK, Loki
## How to debug your K8S workloads?

<table>
<thead>
<tr>
<th></th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stackdriver/Cloudwatch logs</td>
<td>Easy to use&lt;br&gt;Low maintainence efforts</td>
<td>Vendor lock-in&lt;br&gt;High cost&lt;br&gt;Public cloud only</td>
</tr>
<tr>
<td>Splunk/Sumo Logic/Datadog</td>
<td>Easy to use&lt;br&gt;Low maintainence efforts</td>
<td>Vendor lock-in&lt;br&gt;High cost</td>
</tr>
<tr>
<td>EFK</td>
<td>The most popular and mature OSS logging solution</td>
<td>High resource footprint</td>
</tr>
<tr>
<td>Grafana loki</td>
<td>Emerging star&lt;br&gt;Low cost</td>
<td>Slower log content search&lt;br&gt;Need time to become mature</td>
</tr>
</tbody>
</table>
K8S common logging architecture

- Logging Agent
- Logging Backend
- Logging Console
K8S distribution’s logging requirements

CNCF: A K8S distribution can be installed to public(or private) cloud or bare mental

Only OSS logging solutions are qualified:
✓ EFK
✓ Loki
Enterprises’ logging requirements

- Maturity
- Access control
- Ability to integrate with existing logging system
- Ability to integrate with big data platform
- K8S native
## OSS logging solution analysis

<table>
<thead>
<tr>
<th></th>
<th>Maturity</th>
<th>Access Control</th>
<th>Integration</th>
<th>K8S native</th>
</tr>
</thead>
<tbody>
<tr>
<td>EFK</td>
<td>High</td>
<td>Medium</td>
<td>Good</td>
<td>Poor</td>
</tr>
<tr>
<td>Loki</td>
<td>Low</td>
<td>Good</td>
<td>Poor</td>
<td>Good</td>
</tr>
</tbody>
</table>
K8S distribution + Enterprise users + Logging = ?
Logging solution of KubeSphere

Agent: Fluent Bit instead of Fluentd
Backend: Elasticsearch
Console: Customized logging console
Integration: Kafka, Fluentd
K8S native: Search by ns, workload…
Multi-Tenant architecture

3-Tier RBAC based multi-tenant architecture

Cluster: admin/regular/workspace manager

Workspace: admin/regular/viewer

Project: admin/operator/viewer
Multi-Tenant logging APIs

/kapis/logging.kubesphere.io/v1alpha2/cluster
/kapis/logging.kubesphere.io/v1alpha2/workspaces/{workspace}
/kapis/logging.kubesphere.io/v1alpha2/namespaces/{namespace}
/kapis/logging.kubesphere.io/v1alpha2/namespaces/{namespace}/workloads/{workload}
/kapis/logging.kubesphere.io/v1alpha2/namespaces/{namespace}/pods/{pod}
/kapis/logging.kubesphere.io/v1alpha2/namespaces/{namespace}/pods/{pod}/containers/{container}
Multi-Tenant benefits for logging

- User can only access logs he is authorized to
- Cluster admin manages log setting like:
  - Where to send the logs
  - Which logs should be collected
  - When to send the logs
Why Fluent Bit?

Both projects share a lot of similarities, Fluent Bit is fully based on the design and experience of Fluentd architecture and general design. Choosing which one to use depends on the final needs, from an architecture perspective we can consider:

- Fluentd is a log collector, processor, and aggregator.
- Fluent Bit is a log collector and processor (it doesn’t have strong aggregation features like Fluentd).

The following table describes a comparison in different areas of the projects:

<table>
<thead>
<tr>
<th></th>
<th>Fluentd</th>
<th>Fluent Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope</td>
<td>Containers / Servers</td>
<td>Containers / Servers</td>
</tr>
<tr>
<td>Language</td>
<td>C &amp; Ruby</td>
<td>C</td>
</tr>
<tr>
<td>Memory</td>
<td>~40MB</td>
<td>~450KB</td>
</tr>
<tr>
<td>Performance</td>
<td>High Performance</td>
<td>High Performance</td>
</tr>
<tr>
<td>Dependencies</td>
<td>Built as a Ruby Gem, it requires a certain number of gems.</td>
<td>Zero dependencies, unless some special plugin requires them.</td>
</tr>
<tr>
<td>Plugins</td>
<td>More than 650 plugins available</td>
<td>Around 35 plugins available</td>
</tr>
<tr>
<td>License</td>
<td>Apache License v2.0</td>
<td>Apache License v2.0</td>
</tr>
</tbody>
</table>

Consider Fluentd mainly as an Aggregator and Fluent Bit as a Log Forwarder, we can see both projects complement each other providing a full reliable solution.

- [https://docs.fluentbit.io/manual/about/fluentd_and_fluentbit](https://docs.fluentbit.io/manual/about/fluentd_and_fluentbit)
- Resource footprint: 40MB vs 450KB
- Efficiency: C & Ruby vs C
- Dependencies: Ruby gems vs Zero
- Plugins: 650 vs 35
Why Fluent Bit?

**Aggregator vs Collector**
Introducing FluentBit Operator

FluentBit can not reload config gracefully

https://github.com/fluent/fluent-bit/pull/842
https://github.com/fluent/fluent-bit/issues/365

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 "HIP" 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 & DBSync.

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).
Introducing FluentBit Operator

Why FluentBit Operator?
- Config output, filter from UI
- Standard CRD process
- kubectl edit fluentbit fluent-bit
- Reload FluentBit without pod restarting
Introducing FluentBit Operator

How to reload FluentBit config gracefully?

- Configmap reload sidecar
  https://github.com/jimmidyson/configmap-reload

- Customized reload interface:
  http://localhost:24444/api/config.reload
Introducing KubeSphere logging console

K8S log config backed by FluentBit Operator
# Introducing KubeSphere logging console

KubeSphere provides a native log search console for Kubernetes. This console allows you to search logs directly from the K8S cluster, making it easier to troubleshoot issues.

## Log Search

The console includes filters for Project, Workload, Pod, Container, and Key Word. You can search logs by entering conditions and keywords.

### Search Result

<table>
<thead>
<tr>
<th>Time</th>
<th>Project</th>
<th>Pod</th>
<th>Container</th>
<th>Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-06-20 16:04:21</td>
<td>kubesphere-monitoring-system</td>
<td>prometheus-k8s-system-0</td>
<td>prometheus</td>
<td>level=error ts=2019-06-20T08:04:21.741334212Z caller=endpoints.go:130 component=&quot;discovery manager scrape&quot; discovery=k8s role=endpoint msg=&quot;endpoints informer unable to sync cache&quot;</td>
</tr>
<tr>
<td>2019-06-20 16:04:21</td>
<td>kubesphere-monitoring-system</td>
<td>prometheus-k8s-system-0</td>
<td>prometheus</td>
<td>level=error ts=2019-06-20T08:04:21.7414565018Z caller=endpoints.go:130 component=&quot;discovery manager scrape&quot; discovery=k8s role=endpoint msg=&quot;endpoints informer unable to sync cache&quot;</td>
</tr>
<tr>
<td>2019-06-20 16:04:21</td>
<td>kubesphere-monitoring-system</td>
<td>prometheus-k8s-system-0</td>
<td>prometheus</td>
<td>level=error ts=2019-06-20T08:04:21.7416135022Z caller=endpoints.go:130 component=&quot;discovery manager scrape&quot; discovery=k8s role=endpoint msg=&quot;endpoints informer unable to sync cache&quot;</td>
</tr>
<tr>
<td>2019-06-20 16:04:21</td>
<td>kubesphere-monitoring-system</td>
<td>prometheus-k8s-system-0</td>
<td>prometheus</td>
<td>level=error ts=2019-06-20T08:04:21.741898252Z caller=endpoints.go:130 component=&quot;discovery manager scrape&quot; discovery=k8s role=endpoint msg=&quot;endpoints informer unable to sync cache&quot;</td>
</tr>
<tr>
<td>2019-06-20 16:00:26</td>
<td>kubesphere-monitoring-system</td>
<td>prometheus-k8s-system-0</td>
<td>prometheus</td>
<td>level=error ts=2019-06-20T08:00:26.724582631Z caller=engine.go:532 component=&quot;query engine&quot; msg=&quot;error expanding series set&quot; err=&quot;context deadline exceeded&quot;</td>
</tr>
</tbody>
</table>
Introducing KubeSphere logging console

K8S native log details
Demo
We’re open sourcing

- https://kubesphere.io/
- https://github.com/kubesphere/kubesphere
- https://github.com/kubesphere/fluentbit-operator