Log-Aggregations-systeme im Vergleich

Ein Vergleich zweier Technologiestacks mit unterschiedlichen Log Collectoren.

Überblick

Einblick in die Funktionsweise des **Elastic Stack** sowie von **Grafana Loki** mit den beiden verschiedenen Logsammlern **Promtail** und **Fluentd** gewonnen werden.

Als Visualisierungstools werden **Kibana** und **Grafana** eingesetzt.

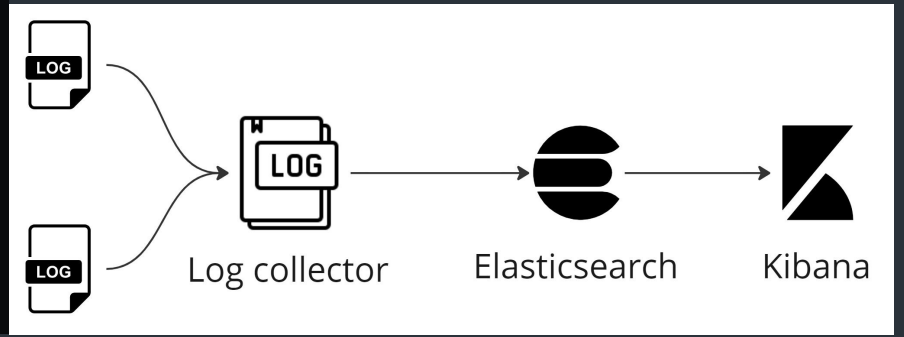
Die Log-Aggregatoren werden mit **Docker Compose** ausgerollt.

Elastic Stack

EFK

- Elasticsearch (Objektspeicher)
- Logstash oder FluentD (Log-Routing und Aggregation)
- Kibana (Visualisierung)

Elasticsearch dient als Objektspeicher und basiert auf Apache Lucene. Es ist in der Lage, unstrukturierte JSON-Objekte zu speichern, wobei sowohl der Schlüssel als auch der Inhalt des Schlüssels indiziert werden.



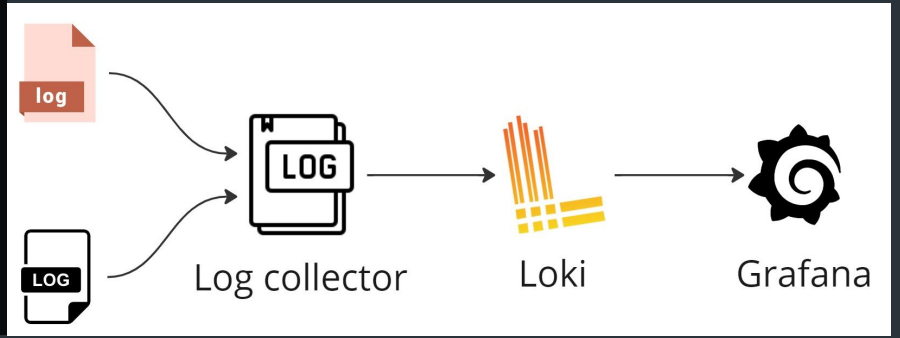
Grafana Loki

Der PLG-Stack, besteht aus Promtail, Loki und Grafana

Das Design von Loki basiert auf dem Vorbild von Prometheus und verwendet einen Index aus Labels, wobei die ursprüngliche Log-Nachricht unindiziert bleibt.

Loki speichert alle Daten in einem einzigen Objektspeicher-Backend wie S3.

Mit Grafana als Dashboard-Tool können Benutzer diese Daten visualisieren und analysieren.



Anforderungen

Anforderungen an den Technologie-Stack für die Protokollierung aus Entwicklersicht.

- **kostenlos**
  kostenlose Tests der wichtigsten Funktionen sind möglich
- **benutzerfreundlich**
  der Technologie-Stack ist einfach zu bedienen und zu konfigurieren
- **schneller und einfacher Rollout**
  auf verschiedenen Systemen möglich - Rollout mit Docker Compose
- **Vollständigkeit**
  alle Logs werden vollständig gesammelt und angezeigt
- **Abfragesprache ist benutzerfreundlich**
  der Technologie-Stack ist einfach zu bedienen und zu konfigurieren
- **intuitives Visualisierungstool**
  die Visualisierungstools Grafana und Kibana sind intuitiv und einfach zu bedienen
- **gut dokumentiert**
  eine ausführliche und verständliche Dokumentation ist verfügbar

Wie findet man heraus, in welchem Umfang die Anforderungen erfüllt sind?

Technology Stacks ausrollen und Dokumentation schreiben.

Dokumentation

How-to in einer Readme auf Github öffentlich zur Verfügung gestellt

Werden alle Logs erfasst und dargestellt?

Anforderung - Vollständigkeit

→ Apache HTTP server benchmarking tool "ApacheBench"

```
ab -n 100 -c 100 http://{Server}:8080/errortest
```

General / Promtail Copy

Last 5 minutes

container | nginx-app | Search (case insensitive) | Enter variable value | job | management | stream | stderr

Total Count of logs - app

200

Total Count of logs - management

10675

nginx-app logs in bytes

{container="nginx-app", job="app", logstream="stderr"}

24 kB

{container="nginx-app", job="app", logstream="stdout"}

11 kB

Panel Title

Dashboard created by Sarah Mai (hope you like it!)

Live logs with searchable_pattern - - in nginx-app

Common labels: nginx-app app

```
> 2023-01-19 15:57:49 stdout 10.208.215.250 - - [19/Jan/2023:14:57:49 +0000] "GET /errortest HTTP/1.0" 404 153 "-" "ApacheBench/2.3" "-"
> 2023-01-19 15:57:49 stdout 10.208.215.250 - - [19/Jan/2023:14:57:49 +0000] "GET /errortest HTTP/1.0" 404 153 "-" "ApacheBench/2.3" "-"
> 2023-01-19 15:57:49 stderr 2023/01/19 14:57:49 [error] 30#30: *100 open() "/usr/share/nginx/html/errortest" failed (2: No such file or directory), client: 10.208.215.25
  0, server: localhost, request: "GET /errortest HTTP/1.0", host: "deneb01.codefactory.levigo.de:8080"
> 2023-01-19 15:57:49 stderr 2023/01/19 14:57:49 [error] 30#30: *99 open() "/usr/share/nginx/html/errortest" failed (2: No such file or directory), client: 10.208.215.250,
  server: localhost, request: "GET /errortest HTTP/1.0", host: "deneb01.codefactory.levigo.de:8080"
> 2023-01-19 15:57:49 stderr 2023/01/19 14:57:49 [error] 30#30: *82 open() "/usr/share/nginx/html/errortest" failed (2: No such file or directory), client: 10.208.215.250,
  server: localhost, request: "GET /errortest HTTP/1.0", host: "deneb01.codefactory.levigo.de:8080"
> 2023-01-19 15:57:49 stderr 2023/01/19 14:57:49 [error] 30#30: *81 open() "/usr/share/nginx/html/errortest" failed (2: No such file or directory), client: 10.208.215.250,
  server: localhost, request: "GET /errortest HTTP/1.0", host: "deneb01.codefactory.levigo.de:8080"
> 2023-01-19 15:57:49 stderr 2023/01/19 14:57:49 [error] 30#30: *80 open() "/usr/share/nginx/html/errortest" failed (2: No such file or directory), client: 10.208.215.250,
  server: localhost, request: "GET /errortest HTTP/1.0", host: "deneb01.codefactory.levigo.de:8080"
```

Total Count of live logs

200

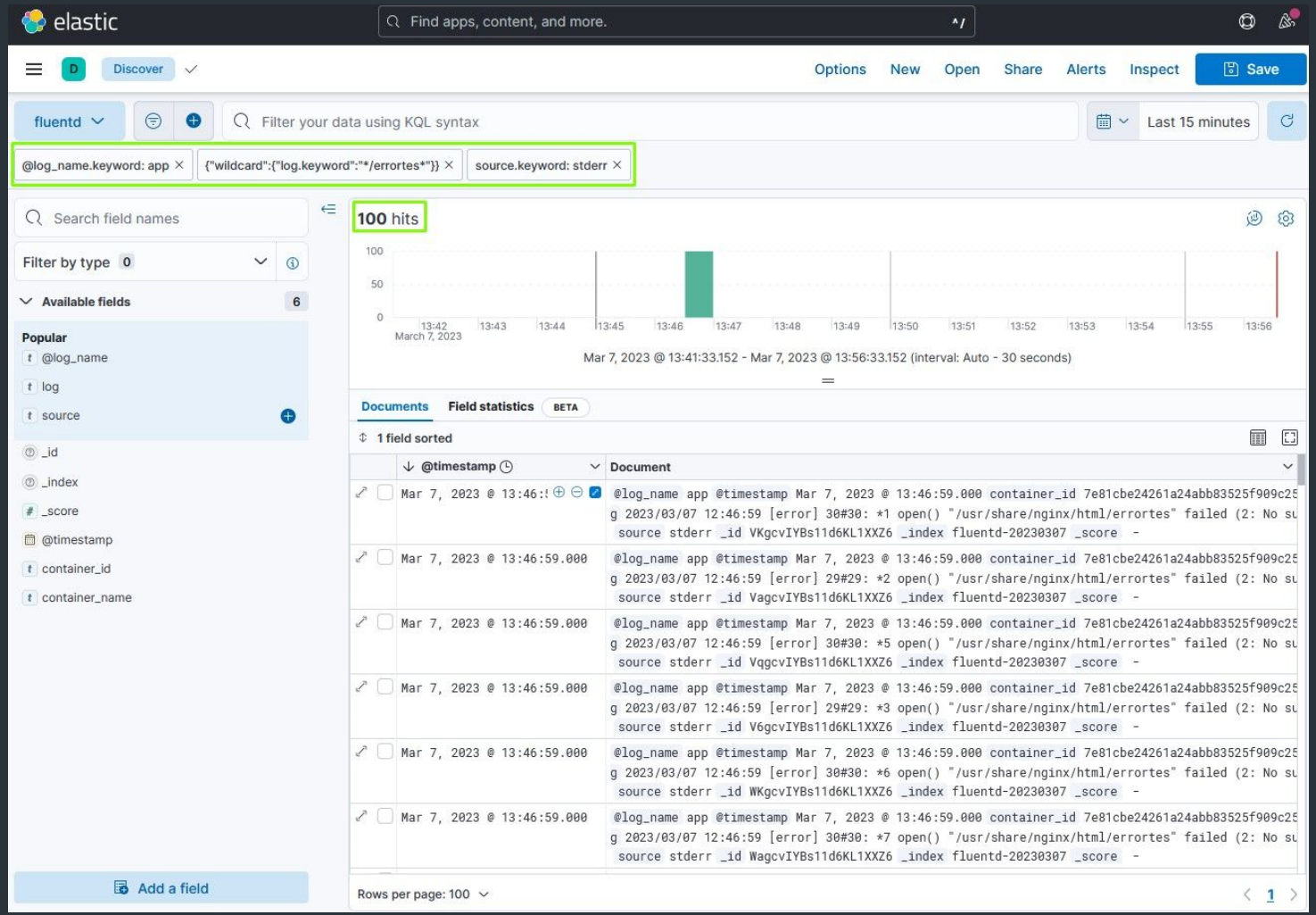
stderr historical

Total count of stderr

	Value
{container="loki"}	10522
{container="nginx-app"}	100
{container="promtail"}	1

Total count of log_stream in container "nginx-app"

	Value
{logstream="stderr"}	100
{logstream="stdout"}	100


elastic
Find apps, content, and more.
Discover
Options
New
Open
Share
Alerts
Inspect
Save
fluentd
Filter your data using KQL syntax
Last 15 minutes
@log_name.keyword: app
{"wildcard":{"log.keyword":"*/errortes*"}}
source.keyword: stderr
Search field names
Filter by type 0
Available fields 6
Popular
@log_name
log
source
_id
_index
_score
@timestamp
container_id
container_name
Add a field
100 hits
100
50
0
13:42
March 7, 2023
13:43
13:44
13:45
13:46
13:47
13:48
13:49
13:50
13:51
13:52
13:53
13:54
13:55
13:56
Mar 7, 2023 @ 13:41:33.152 - Mar 7, 2023 @ 13:56:33.152 (interval: Auto - 30 seconds)
Documents
Field statistics
BETA
1 field sorted
@timestamp
Document
Mar 7, 2023 @ 13:46:59.000
@log_name app @timestamp Mar 7, 2023 @ 13:46:59.000 container_id 7e81cbe24261a24abb83525f909c25
g 2023/03/07 12:46:59 [error] 30#30: *1 open() "/usr/share/nginx/html/errortes" failed (2: No su
source stderr _id VKgcvIYBs11d6KL1XXZ6 _index fluentd-20230307 _score -
@log_name app @timestamp Mar 7, 2023 @ 13:46:59.000 container_id 7e81cbe24261a24abb83525f909c25
g 2023/03/07 12:46:59 [error] 29#29: *2 open() "/usr/share/nginx/html/errortes" failed (2: No su
source stderr _id VagcvIYBs11d6KL1XXZ6 _index fluentd-20230307 _score -
@log_name app @timestamp Mar 7, 2023 @ 13:46:59.000 container_id 7e81cbe24261a24abb83525f909c25
g 2023/03/07 12:46:59 [error] 30#30: *5 open() "/usr/share/nginx/html/errortes" failed (2: No su
source stderr _id VqgcvIYBs11d6KL1XXZ6 _index fluentd-20230307 _score -
@log_name app @timestamp Mar 7, 2023 @ 13:46:59.000 container_id 7e81cbe24261a24abb83525f909c25
g 2023/03/07 12:46:59 [error] 29#29: *3 open() "/usr/share/nginx/html/errortes" failed (2: No su
source stderr _id V6gcvIYBs11d6KL1XXZ6 _index fluentd-20230307 _score -
@log_name app @timestamp Mar 7, 2023 @ 13:46:59.000 container_id 7e81cbe24261a24abb83525f909c25
g 2023/03/07 12:46:59 [error] 30#30: *6 open() "/usr/share/nginx/html/errortes" failed (2: No su
source stderr _id WKgcvIYBs11d6KL1XXZ6 _index fluentd-20230307 _score -
@log_name app @timestamp Mar 7, 2023 @ 13:46:59.000 container_id 7e81cbe24261a24abb83525f909c25
g 2023/03/07 12:46:59 [error] 30#30: *7 open() "/usr/share/nginx/html/errortes" failed (2: No su
source stderr _id WagcvIYBs11d6KL1XXZ6 _index fluentd-20230307 _score -
Rows per page: 100
1

Auswertung der Anforderungen

Loki mit Promtail

Anforderung	*Wertung*
kostenlos	5
benutzerfreundlich	4
schneller und einfacher Rollout	5
Vollständigkeit	5
Abfragesprache ist benutzerfreundlich	5
intuitives Visualisierungstool	4
gut dokumentiert	5

33

Loki mit Fluentd

Anforderung	*Wertung*
kostenlos	5
benutzerfreundlich	3
schneller und einfacher Rollout	5
Vollständigkeit	1
Abfragesprache ist benutzerfreundlich	5
intuitives Visualisierungstool	4
gut dokumentiert	3

26

Elastic mit Fluentd

Anforderung	*Wertung*
kostenlos	3
benutzerfreundlich	3
schneller und einfacher Rollout	5
Vollständigkeit	5
Abfragesprache ist benutzerfreundlich	5
intuitives Visualisierungstool	5
gut dokumentiert	3

29