datagraphis

Logging mit Docker und Elasticsearch

Docker JSON File Logging Driver mit Filebeat als Docker Container

datagraphis

Mark Holstein

- datagraphis GmbH
- Fachinformatiker - Systemintegration 2002
- PHP Software-Developer 2009
- Symfony
- Angular
- Konzert-Fotografie
- Ultra-Trail-Lauf

Facebook: Mark Holstein
Twitter: @MarkHolstein3
Instagram: mubuxnet

Themen

- Ausgangssituation
- Klassische Tools für Log-Auswertung
- Lösungen
- Konfiguration
  - Supervisord
  - Docker Build
- Demo
- Q&A

Ausgangssituation

- Microservice-Architektur mit vielen Services
- Verteilte Logfiles
- Keine Auswertung
- Keine Überwachung
- Nur über Konsole erreichbar
- Aufwändig zu konfigurieren und zu verwalten

Klassische Methoden

- Grep
- Tail
- Notepad++
- LogViewer
- Nagios Check

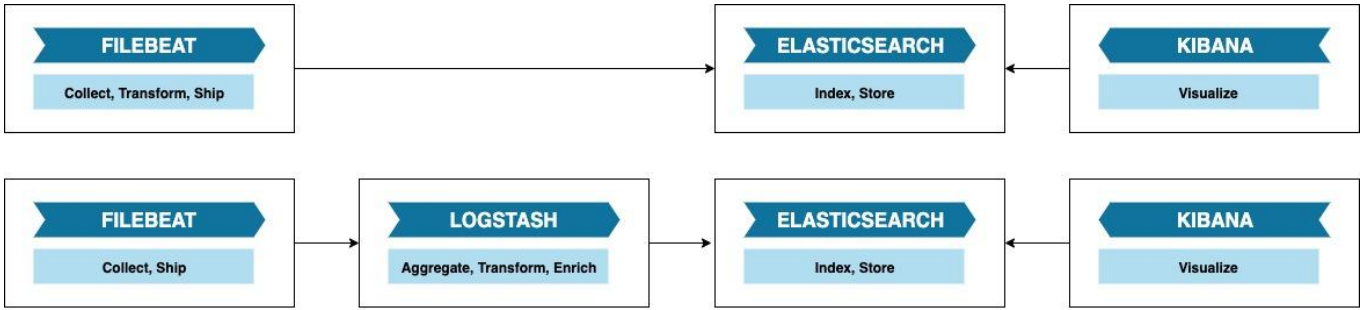
Lösungen

- Splunk
  - Rein kommerzielle Lösung

- Elastic Stack (früher ELK Stack)
  - Elasticsearch, eine Datenbank und Suchmaschine, die Daten (hier: Log-Einträge) skalierbar speichert und indiziert.
  - Kibana, eine UI, um auf die Daten in Elasticsearch bequem zugreifen zu können.
  - Logstash, eine Komponente, um Daten einzusammeln, transformieren und anzureichern.
  - Beats, ein Daten-Collector. Es gibt mit Filebeat, Metricbeat, Packetbeat etc. verschiedene spezialisierte Implementierungen der Beats-API.

Logstash oder Beats?

- Logstash
  - gilt als ressourcenintensiv
  - Sollte nicht mehr als Daten-Collector eingesetzt werden
- Beats
  - Neu-Entwicklung
  - Ausschließlich zum Daten sammeln und Weiterleiten
  - unterstützt einfache Transformationen (z. B. JSON Decoding)

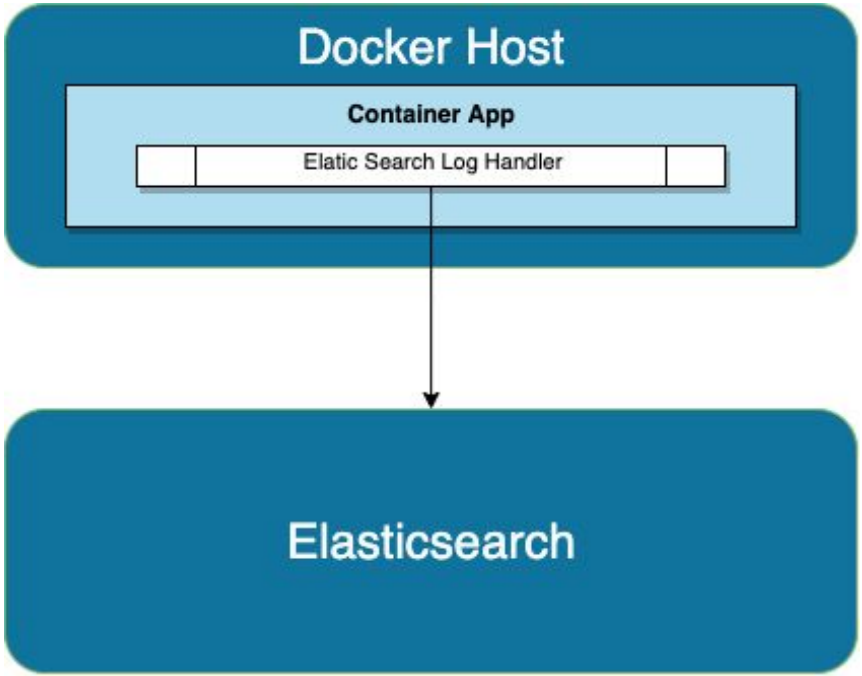


Elastic Stack, mit und ohne Logstash

Log-Handler

Log-Events werden asynchron (d. h. in einem eigenen Thread) an Elasticsearch gesendet

- Laufzeitabhängigkeit zum Log-Aggregator.
  - Wenn Elasticsearch nicht erreichbar ist, läuft irgendwann die Backlog Queue voll.
- Der Handler ist Teil der Anwendung.
  - Bei Herunterfahren oder Crash der Anwendung gibt es keine Log-Einträge
- Ausgaben erst ab Start des Handler
  - Ausgaben vor dem Start, die ggf. Fehlerursachen, werden nicht verarbeitet.
- Konfiguration der Elasticsearch Anbindung erfolgt direkt in der Anwendung.
  - Durch diese enge Kopplung müssen bei einer Konfigurationsänderung alle Container neu deployed werden.



Ein entsprechender Handler ist also für die Produktion in der Regel nicht zu empfehlen

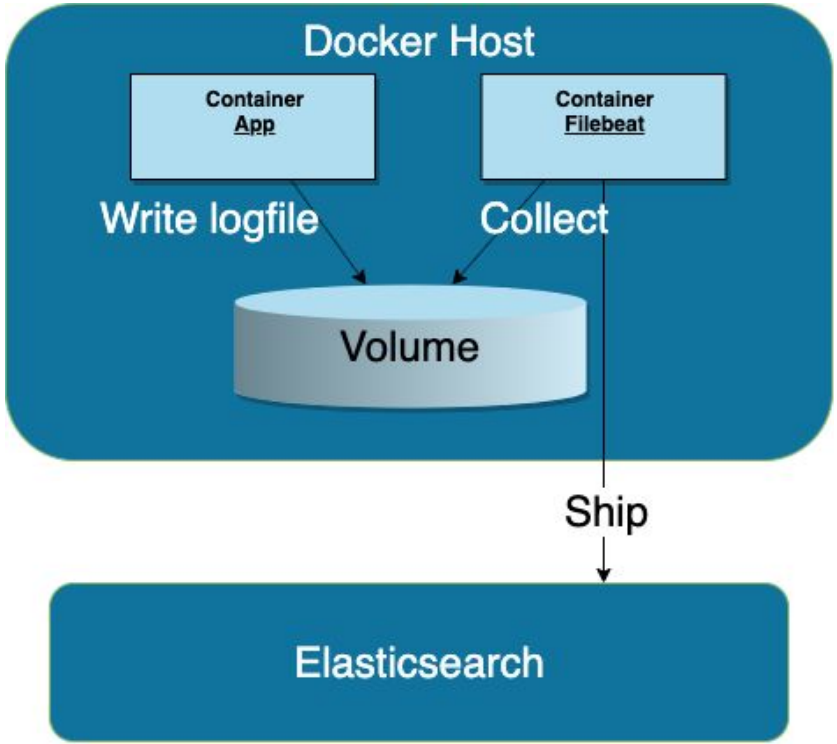
Sidecar Container mit Shared Volume

Die Log-Verarbeitung wird mit Filebeat in einem zweiten Container (sog. Sidecar-Container) ausgeführt.

- Informationen zur Laufzeitumgebung fehlen
  - Wie Host, Instance-ID, Image-Name, Image-Version und Docker-Version des Anwendungscontainers
- Keine Logs der Docker-Engine
  - z. B. dass ein Container nicht gestartet werden konnte.
- Konfiguration des Loggings direkt in der Anwendung
  - wie File-Appender und Log-Rotation

Variante:

- Mehrere Docker Container können in ein Volume schreiben
  - Unterschiedliche Log-Pfade
  - Name der Komponente Teil der Log-Eintrags



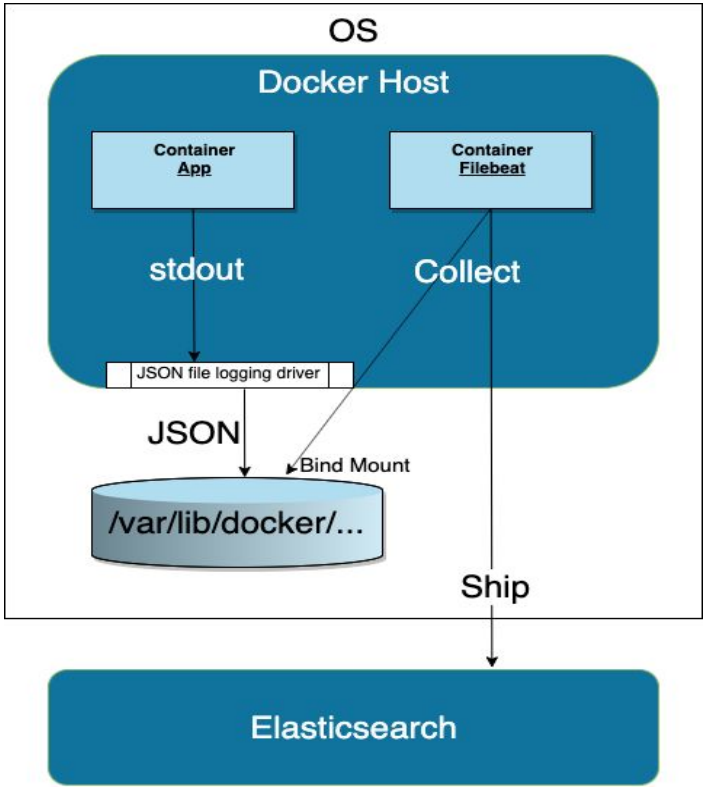
Logging mit Docker und Elasticsearch

datagraphis

Docker JSON File Logging Driver mit Filebeats als Docker Container

Docker loggt standardmäßig über den JSON File Logging Driver alle Ausgaben von `stdout` und `stderr` eines Containers in eine Log-Datei.

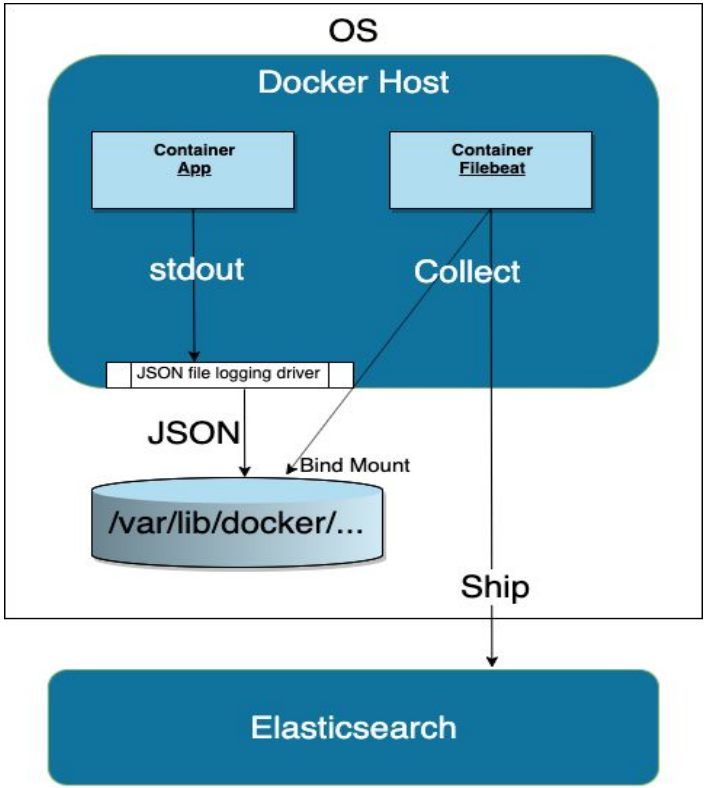
- Anwendung loggt nur nach `stdout` und damit ins docker log
- Filebeat sammelt diese Logs ein, reichert sie mit Docker-Meta-Daten an und schickt sie an Elastic
- Anwendung muss keine Details zur Logging-Architektur kennen



Docker JSON File Logging Driver mit Filebeats als Docker Container

Zu beachten

- Registry Datei sollte in einem Volume persistiert werden
  - Enthält Daten bereits verarbeiteter Logs
  - Um bei Container Rebuild nicht alle Log-Files erneut zu verarbeiten
- Filebeat benötigt entsprechende Zugriffsrechte
- Endlosschleife bei Fehler möglich
  - Filebeat loggt nach stderr. Wenn es zu einem Fehler dabei kommt, kann das zu einer Endlosschleife führen.
  - Um das zu verhindern kann Filebeat in eine Datei loggen.



Config

Dockerfile

- ```
  # Special link to get Output from cronjobs to stdout
  RUN ln -sf /proc/1/fd/1 /var/log/stdout.log
  ```
  - <script> > /var/log/stdout.log

Supervisord

- ```
  [supervisord]
  nodaemon=true
  logfile=/dev/null
  logfile_maxbytes=0
  stdout_logfile=/dev/stdout
  stdout_logfile_maxbytes=0
  redirect_stderr=true
  ```