Bachelorarbeit

Umsetzung von Logging-Richtlinien und Einrichtung eines zentralisierten Logging-Servers für das CFT Portale der Kassenärztlichen Vereinigung Westfalen-Lippe

Kevin Bollich
geboren am 19.04.1997
Matr.-Nr.: 7102160

An der Fachhochschule Dortmund im Fachbereich Informatik erstellte Bachelorarbeit
im Studiengang Software- und Systemtechnik Dual - Vertiefungsrichtung Softwaretechnik

zur Erlangung des akademischen Grades
Bachelor of Science
B. Sc.

Betreuung durch:
Prof. Dr. Martin Hirsch

12. November 2020

Inhaltsverzeichnis

Abbildungsverzeichnis

1. Einführung

1.1. Motivation

Das CFT Portale der Kassenärztlichen Vereinigung Westfalen-Lippe (KVWL) verwaltet eine hohe Anzahl an Applikationen, bei denen regelmäßig neue Funktionen hinzukommen. Bei der stetigen Weiterentwicklung können während der Laufzeit Fehler auftreten, deren Herkunft nicht immer eindeutig ist. Damit die Herkunft solcher Fehler erkannt werden kann, sollten bestimmte Laufzeitinformationen geloggt werden. Da derzeit keine klare Struktur im Logging erkennbar ist, ist das Bugtracking im CFT Portale sehr zeitaufwendig. Der Grund dafür liegt hauptsächlich an der redundanten Serververteilung und dem unstrukturierten Logging.

1.2. Problemstellung

Im CFT Portale müssen die Entwickler regelmäßig die Ursachen von aufgetretenen Fehlern analysieren. Dabei sieht der Prozess folgendermaßen aus:
Jede Anwendung ist auf zwei redundanten Servern installiert und schreibt auf dem jeweiligen Server ihre Logs. Damit die Entwickler herausfinden können, wo das entsprechende Log geschrieben wurde, muss auf beiden Servern manuell nach dem Fehler gesucht werden. Da ein Fehler nicht immer sofort nach Auftreten gemeldet wird und die Software trotz des Fehlers weiter läuft, steigt die Menge an geschriebenen Logs. Den Fehler in den Logdateien zu finden, kann durch die fehlende Möglichkeit des Filterns sehr zeitaufwendig werden.
Eine weitere Herausforderung bei der Fehlersuche liegt in den unstrukturierten Informationen in den Logdateien.

Daraus leiten sich folgende Forschungsfragen ab:

- Mit welchem Log-Management-Tool ist ein an die Probleme des CFT Portale angepasstes zentralisiertes Logging möglich?
- Wie kann ein zentralisierter Logging-Server eingerichtet werden?
- Können die Logging-Richtlinien aus der Projektarbeit in der Praxis umgesetzt werden?

1.3. Zielsetzung

Ziel dieser Bachelorarbeit ist es, die in der Projektarbeit definierten Logging-Richtlinien anhand der Software „Vierteljahreserklärung“ durchzuführen. Außerdem soll eine Evaluierung von Log-Management Tools erfolgen, damit herausgefunden werden kann, ob der in der Projektarbeit erwähnte Elastic-Stack die beste Lösung für das CFT Portale ist, um ein zentralisiertes Logging einzurichten. Wenn die Entscheidung über das Log-Management Tool gefunden wurde, soll ein zentralisiertes Logging mit dem Log-Management Tool umgesetzt werden.

1.4. Vorgehensweise

Zu Beginn der Bachelorarbeit erfolgt eine Evaluation von Log-Management-Tools. Mithilfe der Evaluation soll ein passendes Tool identifiziert werden, dass eine effiziente zentralisierte Lösung für das CFT Portale ermöglicht. Bevor dies geschieht, müssen noch die Anforderungen an das Tool aufgestellt werden. Dies geschieht in Absprache mit dem CFT Portale. Nachdem ein Tool identifiziert wurde, soll ein zentralisierter Logging-Server eingerichtet werden. Dieser soll in Zukunft die erstellten Logs sammeln, anzeigen und analysieren. Anschließend sollen in der Applikationen „Vierteljahreserklärung“ alle in der Projektarbeit definierten Richtlinien umgesetzt werden. Zum Schluss wird ein Fazit zum Verlauf der Bachelorarbeit gezogen. Dabei werden die Ergebnisse der Arbeit noch einmal vorgestellt und bewertet.

Vorläufige Gliederung:

- Einführung

- Evaluation von Log-Management Tools
- Einrichten eines zentralisierten Logging-Server
- Umsetzen der Richtlinien
- Fazit

2. Evaluation von Log-Management Tools

In der vorherigen Projektarbeit wurden für das CFT Portale Richtlinien definiert, die in dieser Bachelorarbeit praktisch umgesetzt werden sollen. Eine dieser Richtlinien war die Nutzung von einem zentralisierten Logging-Server mithilfe des Elastic Stack. Jedoch wurden in der Projektarbeit keine weiteren Tools herangezogen, um zu prüfen, ob der Elastic Stack die beste Alternative ist.
In diesem Kapitel werden unterschiedliche Tools, die für das Log-Management genutzt werden können, evaluiert. Das Ziel dieser Evaluation ist zu prüfen, ob es eine bessere Alternative für eine zentralisierte Logging Lösung gibt, als den Elastic Stack. Dafür werden Tools evaluiert, die den kompletten Elastic Stack ersetzen können, aber auch Tools die einzelne Komponenten austauschen können.
Das CFT Portale wäre in der Lage weitere Kosten für ein Tool auf sich zu nehmen, sollte es dem Team die Arbeit erleichtern können. Daher werden Open-Source und Lizenzpflichtige Tools in dieser Evaluation betrachtet. Sollten jedoch zwei Tools gleichermaßen die Anforderungen erfüllen und eines der Tools Open-Source sein, dann wird sich für das Open-Source Tool entschieden, um kosten zu sparen.
Damit eine Evaluation erfolgen kann, müssen Anforderungen aufgestellt werden. Die Anforderungen sollen dabei helfen eine Entscheidung bezüglich der Tools treffen zu können. Denn Tools die diese Anforderungen nicht erfüllen können, werden nicht weiter betrachtet. Im nächsten Abschnitt werden die Anforderungen definiert.

2.1. Anforderungen an die Log-Management Tools

In diesem Abschnitt werden Anforderungen für die zentralisierten Logging Tools definiert. Die Anforderungen helfen bei der Entscheidung, ein passendes Tool für

das CFT Portale auszuwählen. Daher wurden in Absprache mit dem Team einige Anforderungen definiert, die das zukünftige Tool haben sollte. Für das CFT Portale spielt Wartung eine wichtige Rolle, daher werden einige Anforderungen sich auf den Wartungsaufwand beziehen.

Da das KV-Netz Sicherheitstechnisch stark abgeschirmt ist, kommt eine Cloud Lösung nicht in Frage. Das bedeutet, dass das Tool eine Selbstorganisierte Lösung bieten muss.
Durch die Menge an Anwendungen die das CFT Portale betreuen muss, ist es wichtig, dass das Tool die einzelnen Logs entweder von den unterschiedlichen Maschinen selbst einsammeln kann oder ein Senden von den Anwendungen heraus möglich ist. Das installieren weiterer Log-Agenten die für das schicken der Logs zuständig wäre sollte vermieden werden. Da sonst weiterer Wartungsaufwand entstehen würde.
Da das CFT Portale sich um keine Datenbanken kümmert ist der aktuelle wissensstand des Teams eher allgemein vorhanden. Denn das Team übernimmt in der Regel die Entwicklung von Provider-hosed Apps im SharePoint Umfeld. Damit das Team also erfolgreich mit den Datenbanken arbeiten kann, müssen Schulungen absolviert werden. Daher ist eine wichtige Anforderung die Speicherung der Logs. Das ausgewählte Tool muss eine Möglichkeit anbieten die Logs zu speichern.
Ein wichtiger Punkt ist die Analyse und Anzeige von Logs. Damit ist eine Oberfläche gemeint, die intuitiv benutzt werden kann, um die Logs anzusehen und zu Analysieren. Jedoch sollte in dem Tool auch die Möglichkeit bestehen, Logs zu Filtern und zu durchsuchen.
Im CFT Portale sind Linux- und Windows Server im Betrieb. Jedoch möchte das Team das Tool gerne auf einer Linux Maschine installieren. Da dort das updaten von neuen Versionen einfacher funktioniert und der wissensstand des Teams da komplett gegeben ist.

Das waren die Vorgaben die Vom CFT Portale für ein Tool definiert wurden. Hier nochmal eine kleine Aufzählung der einzelnen Anforderungen:

- Selbstorganisierte Lösung (Kein CLoud)
- Einsammeln von Logs aus unterschiedlichen Anwendungen

- Speicherung von Logs ohne externe Datenbank
- Anzeige und Analyse von Logs
- Filtern und Dursuchen von Logs
- Installation auf Linux Server

2.2. Log-Management Tools

In der Projektarbeit wurde der Elastic Stack in seiner Funktionsweise und dessen Möglichkeiten schon ausreichend vorgestellt. Daher wird in diesem Kapitel hauptsächlich auf die neu zu evaluieren Tools eingegangen. Alle Tools werden hier einmal vorgestellt mit all ihren Vor- und Nachteilen. Eine Bewertung der Tools wird in Kapitel 2.3 durchgeführt. Bevor die Tools evaluiert werden können, musste eine vorauswahl getroffen werden, um potenzielle Tools finden zu können. Dafür wurden eine Vielzahl an Tools betrachtet und nach den Anforderungen in Kapitel 2.1 ausgewählt.

2.2.1. Graylog

Graylog ist ein Open-Source Log-Management Tool. Dessen Motto ist:

> „less cost, more performance“[Grab]

Das Tool setzt auf Performance. Die Funktionalitäten des Tools beziehen sich auf das Sammeln, verbessern, Speichern und der Analyse von Logs. In Graylog kann man eigene Dashboards erstellen und individuell anpassen. Das Dashboard wird mithilfe von Suchabfragen definiert. Damit nicht jeder Mitarbeiten sich mit den Suchabfragen beschäftigen muss, können die Dashboards untereinander geteilt werden. Graylog bietet zusätzlich vordefinierte Dashboards an, die genutzt werden können.
Mithilfe von Graylog können unmengen an Logs gespeichert werden. Daher ist die Suche in Großen Datenmengen essenziell. In Graylog werden die Logs beim Speichern indiziert, um eine effiziente Suche zu ermöglichen. Die Daten werden beim Speichern geprüft. Bei der Prüfung wird die Struktur genauer untersucht, um festzustellen, ob die Struktur in Ordnung ist. Wenn die Struktur nicht in Ordnung ist,

wird sie verbessert.
Die Architektur von Graylog ermöglicht eine multi-threaded Suche. Jede Suche nutzt dabei mehrere Prozessoren, um möglichst effizient zu sein. Die Suche in Graylog ist dabei einfach aufgebaut. Einfache Boolean Operationen werden für die Suche genutzt. Die dazu benötigten Felder werden durch einfaches klicken ausgewählt. Damit muss keine neue Suchsprache erlernt werden und kann von nicht Fachpersonal genutzt werden. [Grab]

Wenn mehr benötigt wird, als die Open Source Version anbietet, dann kann Graylog als Enterprise Variante gekauft werden. Bei der Enterprise Variante werden Support und zusätzliche Funktionen angeboten. Der genaue Vergleich der beiden Varianten kann in Abbildung A.1 im Anhang A.1 gesehen werden. Zu dem Support gehört Hilfe zu allen Graylog bezogenen Fragen, jedoch bietet Graylog zusätzlich Support für Elasticsearch, MongoDB und Oracle Java SE 8 (oder OpenJDK 8). Sie bieten diesen Support an, weil Graylog diese Produkte benötigt, um laufen zu können. Das bedeutet, dass diese Produkte zusätzlich auf der zu installierenden Maschine installiert werden. Zum Thema der zu installierenden Maschine ist wichtig zu beachten, dass Graylog nur auf Linux-basierten Betriebssystemen installiert werden kann.
Die Graylog Enterprise variante kann bis zu 5 GB/Tag kostenlos erworben werden. Für so kleine Datenmengen ist es daher Ratsam direkt auf die Enterprise Variante zu setzen. [Graa]

Damit Graylog funktionieren kann muss weitere Software auf der Maschine installiert werden. Das bewirkt, dass der Aufwand für das Updaten des Tools zu Problemen führen kann. Beim Betrieb von Graylog muss darauf geachtet werden, dass die Versionen der Tools passen. Das sorgt dafür, dass der Wartungsaufwand sehr Hoch ist.
Abließend können die folgenden Vor- und Nachteile für Graylog zusammengefasst werden:

- Vorteile:

  - Open Source (Enterprise auch möglich)
  - Sehr performant durch multi-threaded Suche und indizierung
  - Speichern großer Datenmengen möglich

- Nachteile:
  - Notwendige Installation von weiterer Software
  - Durch extra Software Updates Problematisch
  - Installation nur auf Linux Maschinen

2.2.2. Sematext

Sematext ist ein kostenpflichtiges Log-Management und Infrastructure Monitoring Tool. Das Tool wird als SaaS (Software as a Service) angeboten, jedoch gibt es zusätzlich eine Enterprise Variante, die das laufen innerhalb der eigenen Infrastruktur ermöglicht. Das heißt, dass eine Kopie der Cloud Version von Sematext auf der eigenen Infrastruktur laufen kann. Sematext unterstützt im Vergleich zu anderen Tools nicht nur das Log-Management, sondern diese Funktionen:

- Infrastructure Monitoring
- Application Performance Management (APM)
- Log Management
- Real User Monitoring

Sematext nutzt für das Log-Management Elasticsearch und Kibana. Das heißt beim Kauf von Sematext erhält man eine fertige Lösung des Elastic Stack. Der Vorteil besteht hier in der Wartung die von Sematext selbst übernommen wird. Ein Vor- und Nachteil besteht beim Log-Shipper, der zusätzlich noch installiert werden muss. Dieser wird nicht von Sematext bestimmt. Daher kann da Flexibel entschieden werden welcher Log Shipper am besten geeignet ist. Jedoch kann dadurch auch nicht sichergestellt werden, dass der Log Shipper mit dem Tool gut funktioniert. mithilfe der Filter Funktion können mit Sematext Benachrichtigungen definiert werden, die

beim auftreten eine Nachricht senden. [Semb]

Damit Sematext in der eigenen Infrastruktur laufen kann muss folgende Software installiert sein:

- Docker
- Kubernetes
- Helm

Sematext Enterprise wird als Helm Chart angeboten. Helm Chart ist ein Paket für Kubernetes. Das Helm Chart beinhaltet alles was nötig ist um Sematext Enterprise nutzen zu können. Da Sematext Enterprise nur als Helm Chart angeboten wird, muss die Installation über Kubernetes erfolgen. [Sema]

2.2.3. Fluentd

Fluentd ist ein Open Source data Collector, der es Anwendern ermöglichen soll, alles zu loggen. Das heißt Fluentd sammelt Daten und leitet sie in gewünschter Form weiter zum Ziel. Also ist Fluentd kein Tool um zentralisiertes Logging zu ermöglichen, sondern nur ein Tool das beim zentralisierten Logging behilflich ist. Mithilfe von Fluentd können Datenströme einheitlich und verständlich ablaufen. Ein Beispiel dafür ist in Abbildung 2.1 zu sehen. Die Datenströme im „Before Fluentd“ Teil sind nicht zu erkennen, wobei mit der Nutzung von Fluentd die Datenströme einheitlich über Fluentd laufen. [Flu]

Fluentd versucht soweit es möglich ist, die erhaltenen Daten zu strukturieren. Dabei werden die Daten im JSON-Format gespeichert. Somit kann Fluentd die Daten einheitlich verarbeiten. Zum Verarbeiten gehören das Sammeln, Filtern, Puffern und die Weitergabe der Logs über verschiedene Quellen und Ziele hinweg. Für weitere Funktionalitäten können Lösungen von der Community genutzt werden. Das wird durch die Steckbare Architektur von Fluentd ermöglicht. Fluentd ist in einer Kombination von C und Ruby entwickelt worden. Deswegen benötigt es nur wenig System Ressourcen um zu Funktionieren. Die standard Version von Fluentd ohne weitere Komponenten benötigt ungefähr 30-40 MB Speicher. [Pro]

Die größten Vorteile von Fluentd sind die Menge an Plugins die verfügbar sind und die Performance. Außerdem verbraucht Fluentd sehr wenig Speicher. Der größte Nachteile ist der Zwang nach Strukturierten Daten, damit ist gemeint, dass es nicht so flexibel möglich ist zu entscheiden wie die Logs auszusehen haben. Fluentd formatiert die erhaltenen Daten immer in JSON-Format.

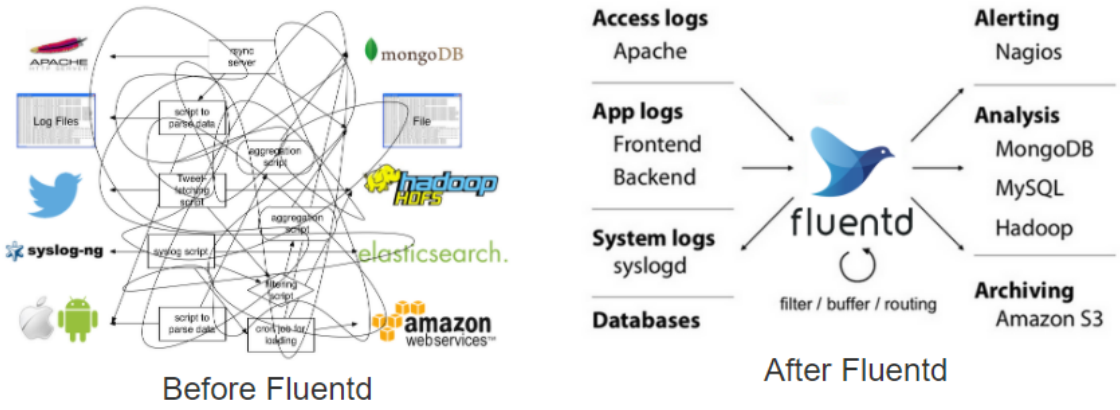

Abbildung 2.1.: Fluentd vorher und nachher Vergleich [Pro]

2.3. Bewertung der Tools

Literatur

[Flu] Fluentd. *Fluentd | Open Source Data Collector | Unified Logging Layer*. URL: https://www.fluentd.org/ (besucht am 10. 11. 2020).

[Graa] Graylog. *Graylog | Open Source vs. Enterprise*. URL: https://www.graylog.org/products/open-source-vs-enterprise (besucht am 04. 11. 2020).

[Grab] Graylog. *Industry Leading Log Management | Graylog*. URL: https://www.graylog.org/ (besucht am 02. 11. 2020).

[Pro] Fluentd Project. *What Is Fluentd? | Fluentd*. URL: https://www.fluentd.org/architecture (besucht am 10. 11. 2020).

[Sema] Sematext. *Sematext Enterprise Overview*. URL: https://sematext.com/docs/sematext-enterprise/ (besucht am 12. 11. 2020).

[Semb] Sematext. *Sematext Enterprise: Log Management & Infrastructure Monitoring Solution*. URL: https://sematext.com/enterprise/ (besucht am 12. 11. 2020).

A. Anhang

A.1. Vergleich Graylog Open Source vs. Enterprise

	OPEN SOURCE Contact sales	GRAYLOG ENTERPRISE Contact sales
Extended log collection using Sidecar	✓	✓
Scalable log collection	✓	✓
Log enrichment data	✓	✓
Simple UI for administration	✓	✓
Graphical log analysis	✓	✓
Content Packs	✓	✓
Alerts & Triggers	✓	✓
REST API	✓	✓
Free marketplace of extensions	✓	✓
LDAP integration	✓	✓
Correlation Engine		✓
Scheduled Reports		✓
Data Forwarder		✓
Offline log Archiving		✓
User Audit Logs		✓
Search Parameters		✓
Technical Support		✓
Search Workflows		✓

Abbildung A.1.: Vergleich Graylog Open Source vs. Enterprise

A.2. Eideststattliche Erklärung

Eideststattliche Erklärung
Ich versichere an Eides statt, dass ich die vorliegende Arbeit selbständig angefertigt und mich keiner fremden Hilfe bedient sowie keine anderen als die angegebenen

Quellen und Hilfsmittel benutzt habe. Alle Stellen, die wörtlich oder sinngemäß veröffentlichten oder nicht veröffentlichten Schriften und anderen Quellen entnommen sind, habe ich als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Dortmund, den 31.08.2020

Kevin Bollich

Erklärung
Mir ist bekannt, dass nach § 156 StGB bzw. § 163 StGB eine falsche Versicherung an Eides Statt bzw. eine fahrlässige falsche Versicherung an Eides Statt mit Freiheitsstrafe bis zu drei Jahren bzw. bis zu einem Jahr oder mit Geldstrafe bestraft werden kann.

Dortmund, den 31.08.2020

Kevin Bollich