OULUN AMMATTIKORKEAKOULU

Joni Arstio

HARMONY RS LOGGING IMPROVEMENT WITH EFK STACK

Centralized logging

HARMONY RS LOGGING IMPROVEMENT WITH EFK STACK

Centralized logging

Joni Arstio
Bachelor's Thesis
Spring 2022
Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Option of Web Development

---

Author(s): Joni Arstio
Title of thesis: Harmony RS Logging Improvement With EFK Stack
Supervisor(s): Lasse Haverinen
Term and year when the thesis was submitted: Spring 2022
Number of pages: 24

Troubleshooting at the current production environments at Topcon Healthcare Solutions EMEA Oy is time consuming and they are implementing new ways to speed up the process. Usually, first place to look for the errors is logging that software and servers provide. It is required to take direct connection to the server that needs actions to access the logs.

Elastic stack provides centralized logging system to overcome with the issue and get logs easily accessible. The objective is to test if Elastic stack would be possible to implement within the current environment and if it is integrable to the company's most widespread product Harmony RS.

Servers used in the project is provided by IBM Cloud with Linux distributions. Libraries and plugins are used to pace up the installations and configurations. Patient privacy is noticed on the process and is filtered in the log stream.

As for result Elastic stack is installed on a Virtual Machine and logs are streaming directly from the software to the Elastic stack server. Fluentd is used to aggregate the log flow and Kibana dashboard will present the data.

The thesis provides the information that Elastic stack can be used to centralize logging within the current stage of the product and in the existing environments. It may not be the method that will be continued with, as there are major upcoming changes in the infrastructure and deploying the products. The basic idea from Elastic stack will remain the same in any case, but the deployment method can be different. This method may not necessarily be the only option if eventually the idea is to unify logging with all products, but it is worth to take into account on Harmony RS future planning.

Keywords: Logs, centralization, monitoring, cloud.

PREFACE

I want to thank Topcon Healthcare Solutions EMEA Oy for the opportunity to implement Elastic stack logging services. There was great demand to centralize logs handing within the current infrastructure, I am glad to be part of the improvements that are going on with application logging and I wish to contribute even more in the future. Thanks to DevOps Architect Janne Rantala for all the help with integrating the services and all the knowledge that helped with the thesis. The instructions you gave will help me a lot when working more intensively with the applications and servers later on. Thank you, Lasse Haverinen, the supervisor of this Bachelor's Thesis from Oulu University of Applied Sciences.

CONTENTS

VOCABULARY

Stack	Collection of software
IBM COS	Company IBM provided Cloud Object Storage
Logstash	Alternative software for log pipeline
Log4j2	Java based logging library
APT repository	Collection of deb packages that can be installed with apt-tools
Apt-tools	Advanced packaging tools to install, update, remove packages
Deb packages	Debian Linux distribution software package
SSL	Secure Sockets Layer
GPG	Gnu Privacy Guard, an encryption technique
PoC	Proof of Concept
MDR	Medical Device Regulation
GDPR	General Data Protection Regulation
FDA	U.S Food and Drug Administration
NMPA	National Medical Products Association
CFDA	China Federal Drug Administration
ARGMD	Australian Regulatory Guidelines for Medical Devices

1 INTRODUCTION

This thesis describes the process of designing and implementing Elastic Stack, centralized logging system. The client of the work is Topcon Healthcare Solutions EMEA Oy which is major supplier of eye healthcare devices and software, together with the parent company Topcon Corporation. The thesis is part of logging improvements and test how Elastic stack would be able to be implemented on the current system and advance into new level with modern ways of log handling.

At the moment services are being monitored, but logs are not easily accessed. To get information from logs, it is needed to connect to the server and search the log files manually. To ease up troubleshooting, centralized logging system will be implemented. End users will get faster responses on the service tickets and systems team will save working hours with faster troubleshooting.

Requirements for the project is to have logs stream from the Harmony RS and its Integration System to a single instance. The service should be easily integrated to customer environments and require minimal changes to the applications. If patient data is present, it should not be visible and stored into storage. It would be good to have the possibility to integrate with other software the company is developing.

With Elastic stack it is possible to aggregate logs from different servers and services. The components in the Elastic stack will collect, store, and visualise the logs data. As dealing with patient data, it is necessary to filter confidential information from the logs.

Harmony Referral System (later Harmony RS) and its Integration Service is the software used for the Elastic stack PoC, as it has the most urgent need for log perusal. It is the product that produce most of the service tickets as it is the key product and most widely used of all Topcon products.

2 SOFTWARE COMPONENTS

The project requires different software to function. All the software used are open-source or free to use so no additional costs ensue. The stack requires server to run on, so indirect costs are incurred. The stack was chosen in advance by the company so comparison and research between providers is not covered. However, there is different methods to implement the stack.

Elastic Stack, also known as ELK Stack, includes four core software. E is for Elasticsearch which stores the logs, L is for Logstash which collects and prepares the logs, and K is for Kibana which lets users search and visualize the log data. Renaming to Elastic Stack was caused by the fourth component Beats being added to the stack in 2015. In this thesis is used a variation of ELK, in which Logstash is replaced with Fluentd. Therefore, the abbreviation for the stack used is EFK [1].

2.1 Elasticsearch

Elasticsearch is the core of the stack which stores all the data centrally and make queries rapidly. It is possible to execute simple queries or combine keywords to get more precise results and its indexing capability makes data accessible faster. Elasticsearch returns queries with numbers, text, geo data, and structured or unstructured data [2].

Elasticsearch is using Apache Lucene search engine to run queries. Lucene is Apache Software Foundation developed software with Java language. Lucene is fast, powerful, and flexible software that can update and search simultaneously. It can run various query types like phrase queries, wildcard queries, proximity queries and range queries [3].

In this project Elasticsearch is installed in IBM virtual server. Logs are being forwarded to Elasticsearch and the data is saved in IBM COS. Cloud Object Storage is the disposal of all data that is processed with Elasticsearch. COS offers buckets for storing objects that can be accessed with HTTP URL.

2.2 Fluentd

Fluentd is lightweight and easy to set up log collecting and forwarding tool written in Ruby. Fluentd does most of the work as it will process all the data that is being sent to Elasticsearch. Fluentd can aggregate log data from different streams and services and pass the data formatted or filtered to Elasticsearch. Filtering the confidential information will be done by Fluentd using regular expressions.

Fluentd is running on the same server with Elasticsearch and receiving the logs sent by the services. Big companies like Amazon and Google offers documentation for Fluentd implementation and can it be integrated with majority of the cloud hosting services or servers. There is great variety of plugins available to make Fluentd fast to set up and configure [4].

Fluentd can be set as forwarder or aggregator. As forwarder it is installed on every instance and pass the data to aggregators. As only one instance is required it will be set up as aggregator.

2.3 Kibana

Visualizing data makes it faster to interpret. Kibana is the software which visualizes data stored in Elasticsearch in the Elastic stack. Kibana supports charts, metrics, tables and graphs through its dashboard. It is possible to create the dashboard needed for each use-case and create reports of any dashboard or visualization screen.

Alerting with predefined rules allows to detect complex conditions and trigger actions when conditions are met. Alerts can be managed in Kibana dashboard. Kibana can detect anomalies and predict future data patterns with its machine learning tools.

3 IMPLEMENTATION

The requirement for the project is to test if Elastic stack is possible to use with the Harmony RS. Elastic stack is predefined to be the arbitrator of the log aggregation and the PoC will be installed on IBM virtual server. The stack should be easily attached to existing customer environments with minimal changes to the application. Patient data must be filtered and not stored in any form. Software for testing is Harmony RS using Java as logging framework. The stack should be able to get the logs directly without log file in between. For future, it would be good to have the possibility to integrate other software too.

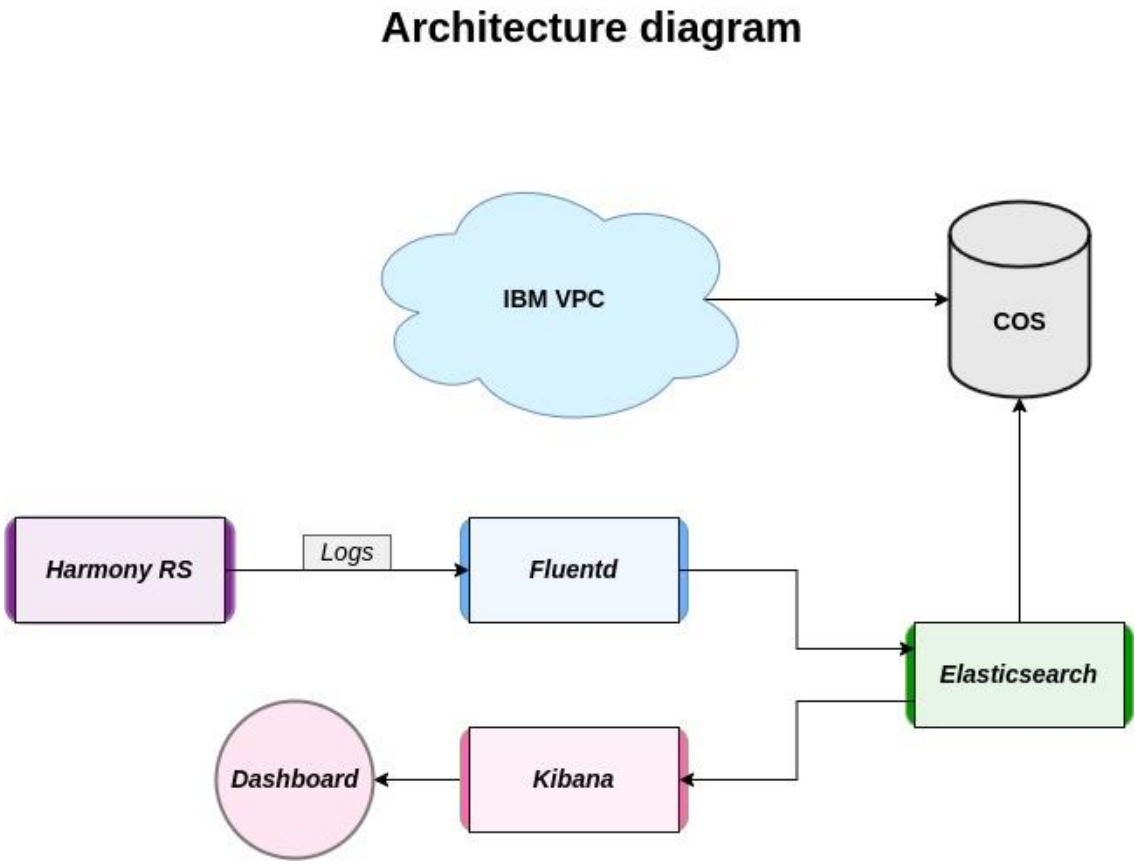

Figure 1: Architecture diagram

IBM Virtual Private Cloud is the cloud provider for the infrastructure. Data is stored on COS and Elasticsearch will be the only component accessing it. Logs are forwarded from Harmony RS via

Fluentd to Elasticsearch, eventually for Kibana to visualize and make it accessible with dashboard as seen in Figure 1.

3.1 Designing

There are different methods building the stack from which components are used to what is the log collection type. As the products are already logging to a file, it would be the easiest way to let Fluentd tail the log files, but to bring logging to modern era, logs are forwarded from the application to Fluentd without the log file in between.

The design process did change on the fly. First option was to use the basic ELK with Logstash being the processing pipeline for data, but as it was known that the services are going to be moved to Kubernetes, Fluentd serves better in that environment because it is more scalable than Logstash and integrates better in multiple type application environments. Deploying Fluentd with Kubernetes is more seamlessly done. Initially it was recommended that no major changes to the software was needed, but with ongoing logging enhancements on the software, it was decided to step up with advanced method to implement the aggregation.

Designing process included communication with Harmony RS architect. Harmony RS integration service needed to be built locally to be able to test the changes to be made as it was not attached to automated building pipelines. Builds are created with Maven which is a tool for Java-based projects by The Apache Software Foundation [5].

As the software updated with Log4j2, it was convenient to take advantage of it. There are libraries designed to use with Log4j2 and Fluentd so it was not necessary to implement it from scratch, instead compare the existing ones.

TABLE 1: Log4j plugins comparison

Library	Pros	Cons	Commits

wycore/ log4j-plugin-fluency	- Based on fluency - Most stars and forks	- One major contributor - Vague company	43 Last 5 years ago
koteswaragoli/ logelastic	- Most commits - MIT-licence	- Not using Fluentd - Documentation - Code editing	60 Last 2 years ago
technologize/ fluency-log4j-appender	- Based on fluency - Newest release - Documentation/examples	- One contributor - Code editing	11 Last 8 months ago

As can be seen in Table 1, Logelastic was most committed and not too aged, but it was not taken into consideration because lack of Fluentd support [6]. Log4j-plugin-fluency is a good competitor but the company behind it was controversial [7]. Fluency-log4j-appender is chosen to proceed with even it requires modification to existing codebase. It provides support for log4j2, is updated most recently and can be implemented on current system [8].

3.2 Environment

The stack requires server to run and as IBM being used already with other services, it was natural to install it into same environment. Virtual machine is set up for minimal resources required and will be scaled up when necessary.

TABLE 2: Pricing of Virtual Machines

Machine type	CPU	RAM	Storage	OS	Cost / hour
Virtual	1	2 GB	100 GB SSD	Debian 10	0.033 €
Virtual	1	4 GB	100 GB SSD	Debian 10	0.040 €
Virtual	2	4 GB	100 GB SSD	Debian 10	0.077 €

Adding just 1 CPU and 2 GB RAM increases hourly costs for over double as seen in Table 2 [9]. The stack does not consume RAM much and 2 GB is enough to begin with. During the testing process Elasticsearch was using from 0.8 GB to 1 GB memory, Kibana under 0.5 GB and Fluentd 0.1 GB. CPU load was under 10% for most of the time. As with only one instance of Harmony RS used in the testing, the cheapest option is enough with 100 GB storage. Changing the server profile can be done with IBM Cloud dashboard and storage can be expanded or made dynamic when more data will be flowing in.

3.3 Installation

The stack is best to use in Linux distributions but supports also Windows Server from 2012 onwards. Previous versions of Elasticsearch and Kibana have stricter requirements of operating system where as Fluentd is better supported overall [10]. Latest version of major release 7 of Elasticsearch and Kibana is used in the project with Fluentd 1.0 [11]. Elasticsearch and Kibana must be using the same version numbers to function properly.

TABLE 3: Operating system support

Operating system	Elasticsearch 7.17.x	Kibana 7.17.x	Fluentd 1.0
Ubuntu 14/16/18/20.04	Yes	From 16.04	Yes
Debian 8/9/10/11	Yes	Yes	Yes
RHEL/CentOS 6/7/8	Yes	From 7.x	Yes
Windows Server 2012/2016/2019/2022	Yes	Yes	Yes

As Table 3 indicates, Debian based distributions has greatest support together with Windows Server releases. Most of the customer environments and servers are using Linux so Windows Server is not taken in account on the project. As Debian being the most used distribution and no immediate changes planned to change it, there is no benefit to go with any other Linux distribution.

3.3.1 Prerequisites

Production environments are based on Linux distribution Debian from versions 8 to 11. To provide integrity, the stack environment is installed with Debian 10. Root privileges are needed to run the commands.

APT transport package is necessary for APT package installation method to be able to download via HTTPS.
Java version 8 or 11 needs to be installed for EFK stack to function. As Java version 11 will be updated in Harmony RS next release, older version of Java will not be used.

3.3.2 EFK

Installing the stack components is straightforward process using the APT repository. It is also possible to install by downloading the Debian package and installing it manually. Manual installation requires more steps than APT and does not grant any advantage, so it is proceeded with APT installation method.

Elasticsearch and Kibana can be installed with less than 10 command-line commands and are preconfigured well by default [12]. As the server running the stack is using Debian Buster Linux distribution, installation package used for Fluentd is called td-agent, which is stable distribution of Fluentd [13]. It is the same Fluentd core but provides easier installation and comes with preconfigured settings.

3.3.3 Configuration

With all the stack components installed it requires configuration for them to communicate. Configuration files are located in:

Elasticsearch: /etc/elasticsearch/elasticsearch.yml
Kibana: /etc/kibana/kibana.yml
Fluentd: /etc/td-agent/td-agent.conf

Elasticsearch has preconfiguration in order to function properly and in this project is used only one instance so default configuration can be used. In the paths section it is possible to change or add multiple locations for storing the data and log files. For now, default values are sufficient. At network tab it is needed to specify default *http.port* to *9200* and restart the Elasticsearch service.

Kibana configuration requires to set *server.port* to *5601* which is default port for Kibana and *elasticsearch.hosts* URL to *[http://localhost:9200].*

Harmony RS Integration Service requires changes for log4j2. It is needed add configuration package for fluency-log4j-appender and forward the log flow to Fluentd [14] [15]. Package is added to log4j2.xml file: <*Configuration packages="io.github.technologize"*>

In this block it is possible to specify names and tags for easier search if multiple instances in use. In this project SSL was not used so the value is set to false. Server host should be the IP of the machine where Elasticsearch is running and set to use port 24224.

Fluentd configuration is added to Appenders block in log4j2.xml file:

```
<Fluentd name="fluentd" tag="Harmony RS Integration Service">
 <Field name="application">Harmony RS Integration Service </Field>
 <FluentdConfig sslEnabled="false">
  <Server host="Elasticsearch IP" port="24224" />
 </FluentdConfig>
</Fluentd>
```

Fluentd is configured to listen port 24224 in which Harmony RS Integration Service logs is sent. Type is set to forward so that logs are streamed directly to Elasticsearch. Other option that can be used is tail and specifying the log file path so Fluentd would read the log file and send it over to Elasticsearch. Binding 0.0.0.0 IP will allow all IP's from local system or anywhere on the internet so for the safety it is better to use Elasticsearch IP.

```
<source>
  @type forward
  port 24224
  bind 0.0.0.0 (Elasticsearch IP)
</source>
```

For testing, Harmony RS Integration Service is installed on a Virtual Box Debian 10 instance. The version was still unreleased 3.8. that included updated Java logging system log4j2. Modifications were made on a standalone version without Harmony RS or other related services installed. Firewall ports needed to be configured for external traffic to be allowed.

To communicate with Elasticsearch the following configuration will match everything Fluentd is processing and pass to Elasticsearch every 10 seconds. Enabling ILM (Index Lifecycle Management) allows automatic configuration of indices with Kibana, e.g., creating new index for each day. Logstash_prefix will create indices names with prefix fluentd. After restarting Fluentd td-agent and selecting the index pattern in the Kibana settings, the logs will show be shown in Kibana dashboard as seen in Figure 2.

```
<match *.**>
  @type elasticsearch
  host 127.0.0.1
  port 9200
  enable_ilm true
  logstash_prefix fluentd
  flush_interval 10s
</match>
```

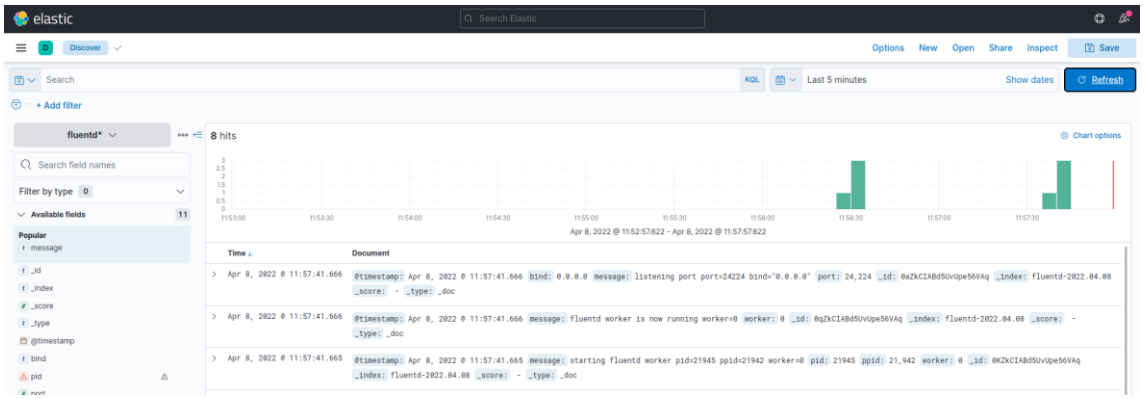

FIGURE 2: Kibana dashboard

In the Fluentd configuration file it is possible to add multiple sources or matches. For example, it would be possible to send virtual machine syslogs to Elasticsearch with describing it in the configuration file. Exposing syslog to port 42185 can be done with adding to /etc/rsyslog.conf line: *.* @127.0.0.1:42185

```
<source>
  @type syslog
  port 42185
  tag syslog
</source>
<match syslog.**>
  @type elasticsearch
  host 127.0.0.1
  port 9200
</match>
```

3.4 Patient privacy

Due to Data Protection Regulation, it is not allowed to collect and store all possible data and in this case any patient data is not important. Only relevant information is what services are sending. To avoid any confidential information stored, there is regular expression built-in plugin at Fluentd that

can be used to filter out any kind of identification data. All the data processed are stored on a well-secured server that cannot be accessed without VPN connection and valid credentials.

As dealing with the medical devices and operating worldwide, there are different regulations between continents. Regulatory responsible for medical devices and operating are listed below.

European Union follows EU MDR (Regulation (EU) 2017/745) on medical devices for human use with GDPR regulation [16].
United States of America's responsible regulatory is FDA's Center for Devices and Radiological Health [17].
People's Republic of China regulatory administration is NMPA (CFDA) [18].
Australia's ARGMD provides the legislative requirements under Department of Health, Therapeutic Good Administration [19].
Middle East do not have common regulatory administration, instead regulated by country [20].

Countries have also their own limitation of data usage. For example, it is not allowed to store data from Germany in the storage that is located in the United Kingdom. Eventually, one stack of Elastic will not be enough, instead it is needed to set up multiple environments according to the continent-wide and country-wide regulations.

3.5 Filtering with Fluentd

Filtering with Fluentd is done at the same configuration file with filter option. At the research of what sensitive information is in the logs, there was userFirstName, userLastName and username like seen in Figure 3 [21], that can be identified to a person. This information is unrelated to any service monitoring that is tried to achieve.

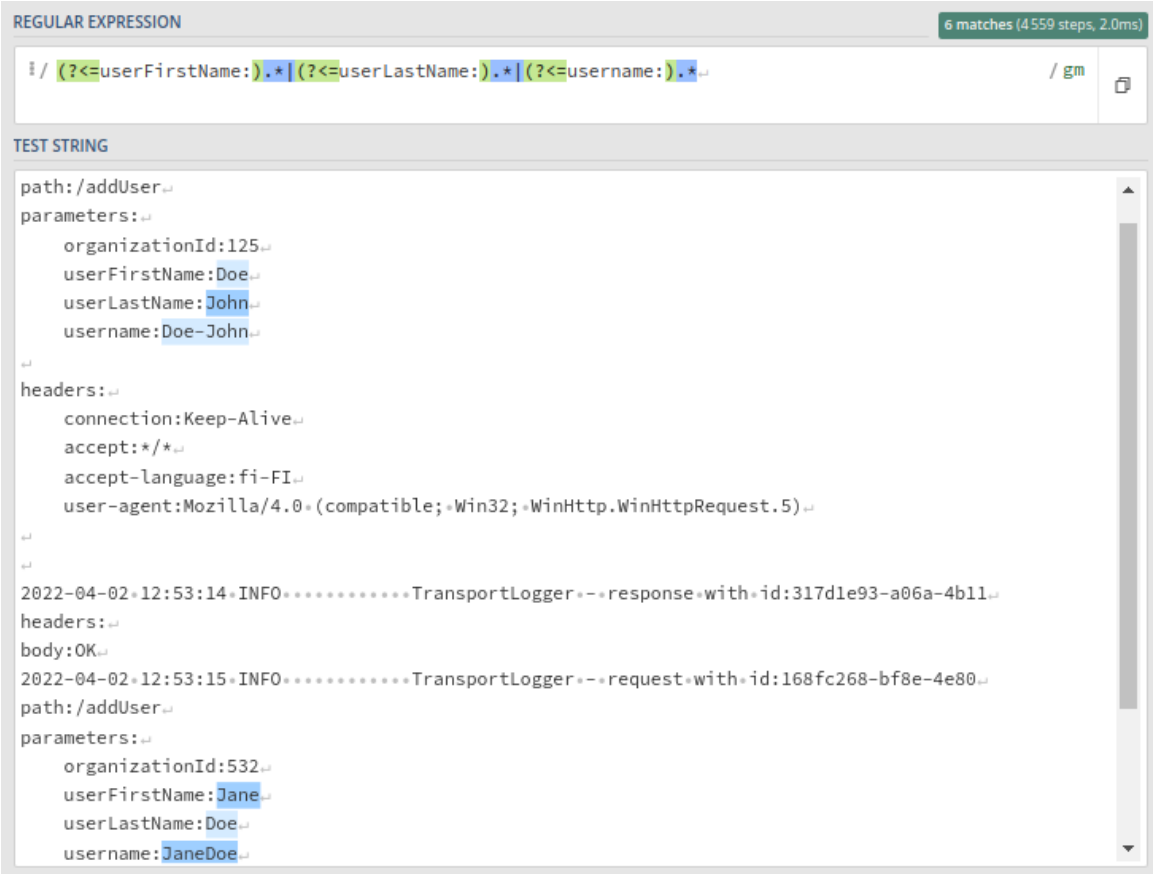

FIGURE 3: Test logs and regular expression

Regular expression positive lookbehind method is used to capture the three strings which value is needed to hide. [21] As the three fields are always the same with the same structure, it is possible to chain the regular expression with pipe character so that all the values are captured with one expression.

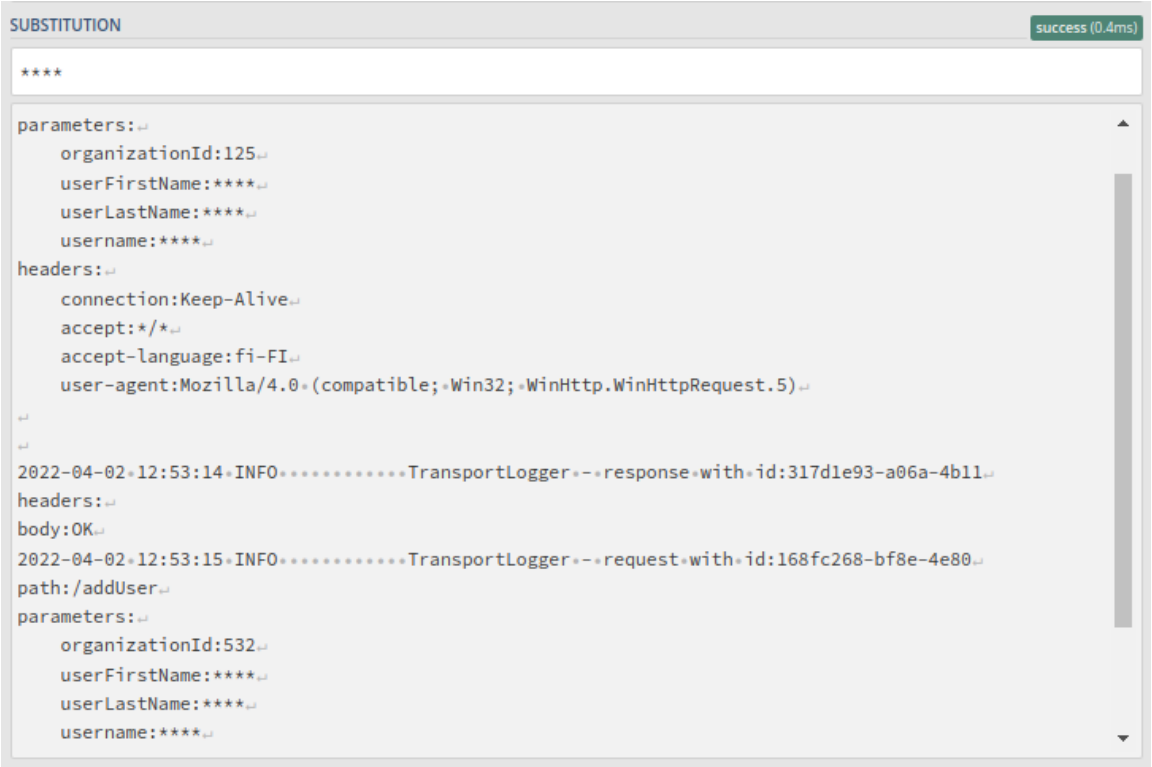

FIGURE 4: Substituting the values

After the value of the sensitive information is captured, it can be substituted with the asterisk characters as seen in Figure 4 [22]. Same thing can be done with Fluentd filter record_transformer option. Record will capture the message with regular expression and substitutes them.

```
<filter>
  @type record_transformer
  <record>
    message ${ record[“message”].gsub(/ (?<=userFirstName:).*|(?<=userLastName:).*|(?<=username:).* /,”****”) }
  </record>
</filter>
```

In this test customer identification data will not be sent over to Elasticsearch. Testing was done with test log that was captured from Harmony RS development environment and setting Fluentd to tail the log file. Harmony RS did not have the support for log4j2 so it could not be included on the testing. Filtering at the end of the log chain will get refactored with the ongoing logging development and this method will become obsolete. For now, it is good to have the option for it if this method will be on the development in near future.

4 FURTHER DEVELOPMENT

The project goal was to help immediate need of log collecting, and part of bigger plan of logging system improvement, and the stack implemented will not necessarily be the exact solution of final logging system. Major upgrades to applications will happen in near future which will require modification to this PoC. Kubernetes implementation will be one of the major changes that requires refactoring.

Next step is to take it to product development pipeline to see the functioning on larger scale. Testing was done in a single instance of Harmony RS Integration Service with the upgraded log4j2 that is not in production yet. Harmony RS can be integrated with the same methods used after getting the log4j2 update. Fluentd is often used with FluentBit which is fast and lightweight logging forwarder that could overcome the problems which might occur when attaching multiple Harmony RS and Integration Services to one Elastic stack.

Kubernetes implementation will change the way logs are handled as a whole. Most common way in Kubernetes is to stream application logs to container stdout and stderr so they can be accessed with Fluentd or set up a sidecar to forward logs to stdout/stderr if application is not doing it. Basic idea of this thesis can be applied to Kubernetes environment, but the deployment will change completely.

This PoC was to test Elastic stack but there are other options also. Azure Application Insights being used on another product the company is developing, but it requires monthly subscription and needs developing on attaching to IBM environment. There have been tests with Splunk and talks about using Prometheus and Grafana instead. If eventually the goal is to get unified logging between all products and services, there are many options to think of. Elastic stack is able to merge them all one way or another.

5 CONCLUSION

The purpose of this thesis was to test if Elastic stack is possible to achieve the goal of centralized logging system within the current environment. After all, Elastic stack is relatively fast to set up and get working. It offers great variety of customization and plugins for all kinds of environments.

As usual, in the software development, the process did change on the fly. ELK became EFK and with high probability will change again before getting to customer environments. This implementation requires changes to Harmony RS and Integration Service codebase which may be blocker when looking further in the future. To get all customer environments attached, it requires updating all the installed Harmony RS and Integration Service instances and it will take months. It would be possible to include this method as part of upcoming version, but the transition period might get too long.

The requirements for this project are met. The PoC is done, and it is working with single instance of Harmony RS Integration Service. The implementation did get delayed for pending log4j2 update that was required to take into account after getting the information of upcoming update. Elastic stack is possible to implement on the updated log4j2 version of Harmony RS and Integration Service and it is flexible if there are updates incoming to the application or the way it sends logs. It does require editing the source code, but this was agreed on before continuing with the updated method. Patient data is possible to filter before storing but advanced way would be to exclude it already at the Harmony RS logs output.

Whole logging system requires refactoring and Elastic stack alone is not able to overcome all the issues that are present. The way software is sending logs needs to be refactored on the root level already, and not fixed at the end of the chain. There are ongoing enhancements for logging system, and it is going to improve step by step.

REFERENCES

[1] Elasticsearch, "Our story," *elastic.co*, Oct. 5, 2018. [Online]. Available: https://www.elastic.co/about/history-of-elasticsearch. [Accessed Feb. 11, 2022].

[2] Elasticsearch, "The heart of the free and open Elastic Stack," *elastic.co*, 2022. [Online]. Available: https://www.elastic.co/elasticsearch. [Accessed Feb. 11, 2022].

[3] The Apache Software Foundation, "Apache Lucene Core," *apache.org*, 2022. [Online]. Available: *https://lucene.apache.org/core.* [Accessed Apr. 14, 2022].

[4] Fluentd, "List of Plugins By Category," *fluentd.org*, 2022. [Online]. Available: https://www.fluentd.org/plugins. [Accessed Feb. 14, 2022].

[5] The Apache Software Foundation, "What is Maven," *apache.org*, Apr. 19, 2022. [Online]. Available: https://maven.apache.org/what-is-maven.html. [Accessed Apr. 04, 2022].

[6] Koteswaragoli, "LogElastic," *github.com*, Nov. 22, 2019. [Online]. Available: https://github.com/koteswaragoli/logelastic. [Accessed Feb. 14, 2022].

[7] Wycore, "log4j-plugin-fluency," github.com, May 12, 2017. [Online]. Available: https://github.com/wycore/log4j-plugin-fluency. [Accessed Feb. 14, 2022].

[8] Technologize, "fluency-log4j-appender," *github.com*, Jul. 26, 2022. [Online]. Available: https://github.com/technologize/fluency-log4j-appender. [Accessed Feb. 14, 2022].

[9] IBM Cloud, "Device Profiles," *ibm.com*, 2022. [Online]. Available: *https://cloud.ibm.com/gen1/infrastructure/virtual-server.* [Accessed Apr. 14, 2022].

[10] Elasticsearch, "Support Matrix," *elastic.co*, 2022. [Online]. Available: https://www.elastic.co/support/matrix#matrix_os. [Accessed Mar. 11, 2022].

[11] Fluentd, "Fluentd (v1.0, current stable)," *fluentd.org*, 2022. [Online]. Available: https://www.fluentd.org/download. [Accessed Mar. 11, 2022].

[12] Elasticsearch, "Install Elasticsearch with Debian Package," *elastic.co, 2022. [Online].* Available: https://www.elastic.co/guide/en/elasticsearch/reference/7.17/deb.html. [Accessed Mar. 15, 2022].

[13] Fluentd, "Install by DEB Package (Debian/Ubuntu)," *fluentd.org*, 2022. [Online]. Available: https://docs.fluentd.org/installation/install-by-deb. [Accessed Mar. 15, 2022].

[14] The Apache Software Foundation, “Using Log4j in Cloud Enabled Applications,” *apache.org*, Feb. 23, 2022. [Online]. Available: https://logging.apache.org/log4j/2.x/manual/cloud.html. [Accessed Mar. 02, 2022].

[15] Paraschiv, Eugen, “How Log4J2 Works: 10 Ways to Get the Most Out Of It,” *stackify.com*, Jun. 14, 2017. [Online]. Available: https://stackify.com/log4j2-java. [Accessed Mar. 02, 2022].

[16] Official Journal of the European Union, “REGULATION (EU) 2017/745 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL,” *europa.eu*, Apr. 05, 2017. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32017R0745. [Accessed Mar. 23, 2022].

[17] U.S. Food & Drug Administration, “Overview of Device Regulation,” *fda.gov*, Apr. 09, 2020. [Online]. Available: https://www.fda.gov/medical-devices/device-advice-comprehensive-regulatory-assistance/overview-device-regulation. [Accessed Mar. 23, 2022].

[18] Tadd, Richard, “What’s in a name? How the CFDA became the NMPA,” *clinicalservicecenter.com*, Feb. 07, 2020. [Online]. Available: https://clinicalservicecenter.com/whats-in-a-name-how-the-cfda-became-the-nmpa. [Accessed Mar. 23, 2022].

[19] Australian Government, Department of Health, Terapeutic Goods Administration, “Australian regulatory guidelines for medical devices (ARGMD),” tga.*gov.au*, Jan. 07, 2022. [Online]. Available: https://www.tga.gov.au/publication/australian-regulatory-guidelines-medical-devices-argmd. [Accessed Mar. 23, 2022].

[20] Samadi, Fatemeh, “Regulatory requirements of Medical Devices in MENA countries,” *dgra.de*, Feb., 2015. [Online]. Available: https://www.dgra.de/media/pdf/studium/masterthesis/master_samadi_f.pdf. [Accessed Mar. 23, 2022].

[21] Goyvaerts, Jan, “Lookahead and Lookbehind Zero-Length Assertions,” *regular-expressions.info*, Jan. 03, 2022. [Online]. Available: https://www.regular-expressions.info/lookaround.html. [Accessed Apr. 04, 2022].

[22] Dib, Firas, “Regex101,” *regex101.com*, Apr. 06, 2022. [Online]. Available: https://regex101.com. [Accessed Apr. 04, 2022].