TEST Engineering & Management

Web Log Collection Monitoring System by using ELK

Seong-Ik Kim[1], Koo-Rack Park[*2], Dong-Hyun Kim[3]
[1,3]Ph. D. Student, Department of Computer Engineering, Kongju National University, 31080, Rep. of Korea.
[*2]Professor, Department of Computer Science & Engineering, Kongju National University, 31080, Rep. of Korea.
jarakag@gmail.com[1], ecgrpark@kongju.ac.kr[*2], dhkim977@naver.com[3]

Article Info
Volume 83
Page Number: 4378 - 4386
Publication Issue:
March - April 2020

Abstract

Establishment and focus: As internet becomes widely spread, users of web and apps are increasing, and even though specification of servers and the network availability are being upgraded to handle a large number of user sessions and traffics in web servers, there are many limitations in solving the problems. Current situation is that the number of servers gets increased to distribute requests of users in order to solve the problem of insufficient capacity to handle sessions and traffics as an alternative solution. Users are requesting the processing of various functions in the webpages, and since all these processes are recorded in the web logs, the amount of the logs is parallel to the number of requests by the users. These logs are frequently used to identify the causes when web issues occur. Since the service administrator has to check logs of each server by accessing a large number of servers one by one in such procedure, it consumes a lot of time resources to check the web logs to identify the web issues.

System: System: In order to reduce the resource, various methods such as FTP (File Transfer Protocol) and Filebeat are used to collect logs of a large number of servers into the central server. In case of using FTP, a user can roll and send logs by designating certain time period or date, and in case of Filebeat, events can be sent through event detection in real-time. But if there is a server of which the transmission is omitted in the process of collecting logs, the reliability of the log analysis data has to drop accordingly. In this study, a plan to enhance the reliability of log analysis data through a system that can monitor whether there is any server of which the collection has been omitted by using the access log analysis data of Apache web server using ELK is proposed. By using a model that is proposed in this study, the time resources that are consumed to collect logs can be saved. It is expected to provide smooth service to users by reducing the time to identify the cause and take necessary measures since integrated logs are checked when the log data needs to be checked due to the occurrence of a failure in future. As for the future study, the study that can establish the process which can automatically take necessary actions through the analysis of the logs. That are recorded during the occurrence of failures shall be continued.

Article History
Article Received: *24 July 2019*
Revised: *12 September 2019*
Accepted: *15 February 2020*
Publication: *26 March 2020*

Keywords: *ELK stack, Elasticsearch, Logstash, Kibana, Beats, Web Log.*

1. Introduction

As much as internet gets spread, the number of users who use the web is increasing. Recently, web apps that are based on HTML5 and Javascript started to be used, and the users who use webs instead of apps are also increasing [1]. Even though specification of servers and the network availability are being upgraded to handle a large amount of user sessions and traffics in web servers, there are limitations. Insufficient capacity to handle sessions and traffics is solved by increasing the number of servers to distribute and process requests of users as an alternative solution. Various UIs that can utilize more information and functions in the webpages are offered. The users make requests to process various functions in the webpages. All these processes are recorded in the web logs, and the volume of the logs is parallel to the number of requests by the users. The number of logs is greatly increasing along with the increase of the number of web users. These logs are frequently used to identify the cause when a web issue occurs, but in this process, the service administrator has to check the logs of each server by accessing a large number of servers one by one. It means that it consumes a lot of resources to analyze web logs including web issues. The log analysis is a process that converts raw logs that have been recorded due to the occurrence of the software system events into information that is useful for the administrator and the manager in solving a problem [2,3]. Such log analysis is used in various domains such as data center operation [4–6] and security threat detection [7,8]. In addition, the function of analyzing logs accurately and quickly can reduce the down time of the system, and it is very important in detecting an operation problem before it occurs or during its occurrence [9]. In order to reduce the resources, various methods such as FTP (File Transfer Protocol) and Filebeat are used to collect logs of a large number of servers into the central server. In case of using FTP, a user can roll and send logs by designating certain time period or date, and in case of Filebeat, events can be sent through the event detection in real-time [10]. But there is a server of which the transmission is omitted in the process of collecting logs, the reliability of the log analysis data has to drop accordingly. In this study, a plan to enhance the reliability of log analysis data through a system that can monitor whether there is any server of which the collection has been omitted by using the access log analysis data of Apache web server using ELK is proposed.

2. Related Works

2.1. Apache Access Log

Apache Access log is one of log files that is a generally used type in the web [11]. Apache provides mod_log_config module, and the type of the access log can be configured using this module [12]. Various information such as the remote host IP, the request processing time, the method, the URI, the protocol, and the HTTP status code can be recorded in the log. Since custom log type can be designated, fields can either be added or removed as needed by a user.

2.2. ELK Stack

The following [Figure 1] is a data flow diagram of Elastic Stack [14].

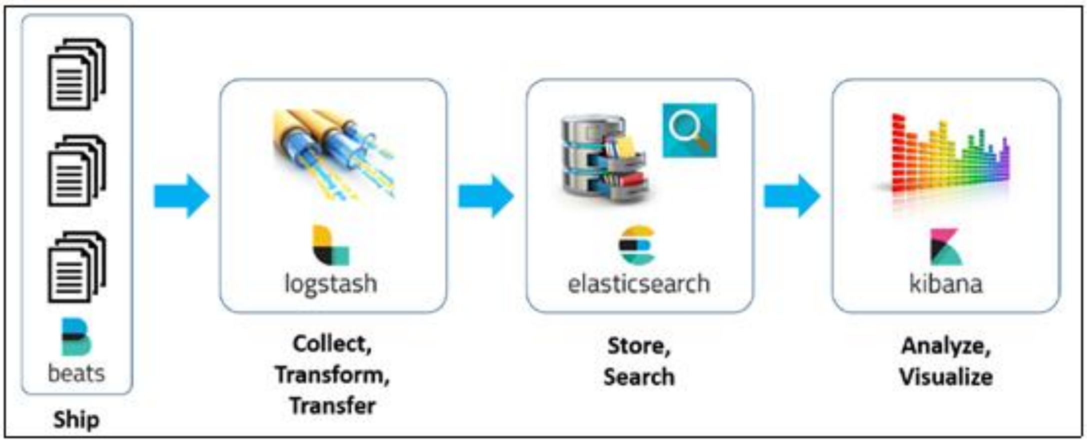


Figure 1. ELK Stack Structure.

After collecting and analyzing the data with the Logstash, it is stored in the Elasticsearch. The data that is stored in the Elasticsearch gets displayed through the Kibana as numbers or graphs so that it is easier for a user to see. ELK Stack is composed of Elasticsearch, Logstash, Kibana, and beats.

The ELK Stack is a collection of software that is specially designed to process, store, query, and visualize the logs [13], which is an integrated solution that is currently called Elastic Stack. The fourth product which is called Beats is a light-weight shipper that sends data from an edge device to Logstash, and it is added to the stack.

Also, the Elastic Stack can be configured as on-premises or it can be used as SaaS (Software as a Service) which is offered by an external company of the cloud.

2.2.1. Elasticsearch

Elasticsearch is a JSON-based document-oriented database, and it is a distributed search and analysis engine [15,16]. The Elasticsearch was implemented based on Apache Lucene, and it is an open source software that is designed to have the most optimal performance. It can be distributed, and it is a search engine with excellent scalability. The inverted index is implemented by using finite state transducers for querying entire texts, BKD tree [17] for storing numerical and geographical data, and the column repository for the analysis. Since the entire text search engine extracts the data that is closely related to the search condition, the Elasticsearch only searches the data that exactly matches the search condition and thus performs the data search more flexibly, compared with the current relational database management system.

Also, the Elasticsearch doesn’t explicitly provide schema for multi logs, but indexes each log data item as a JSON document (simple list of key-value pair). It is designed as a cluster system for the scalability and stability, and additional node can be conveniently used according to the amount of the data and query to be processed. Each document that has been indexed is subdivided as multiple pieces and duplicated copies, and the number can be decided as necessary. If the pieces and duplicated copies are distributed from the node, the stability gets improved. Since RESTful API is provided by using the JSON document through HTTP, the document control, the parameter configuration and the status check can be conveniently performed through the HTTP interface of the external application program [14]. Such Elasticsearch can provide a help to the real-time big data analysis through the combination of high flexibility and convenient expansion options for each requirement [18].

The following [Figure 2] is a configuration diagram of Elasticsearch cluster.

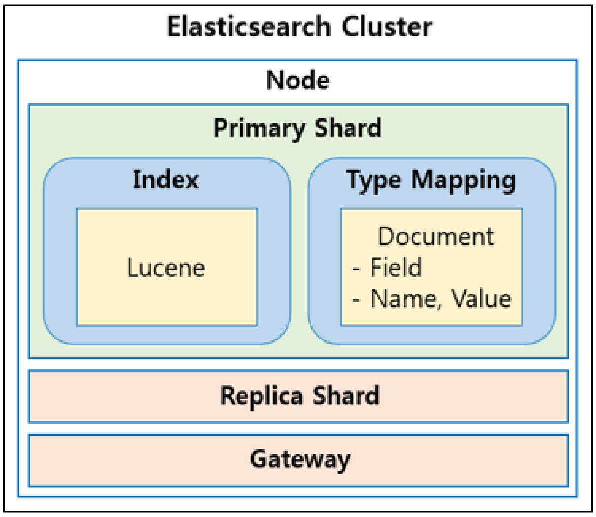


Figure 2. Elasticsearch Cluster Configuration.

2.2.2. Logstash

Logstash is a message processing pipeline, and a message is processed in the steps of input, filter, and output [14]. It is an open-source data collecting tool that has various and flexible functions and transmission pipelines that can process and display all logs that can be created in web servers, systems, applications, and error logs. The following [Figure 3] is the Logstash structure.

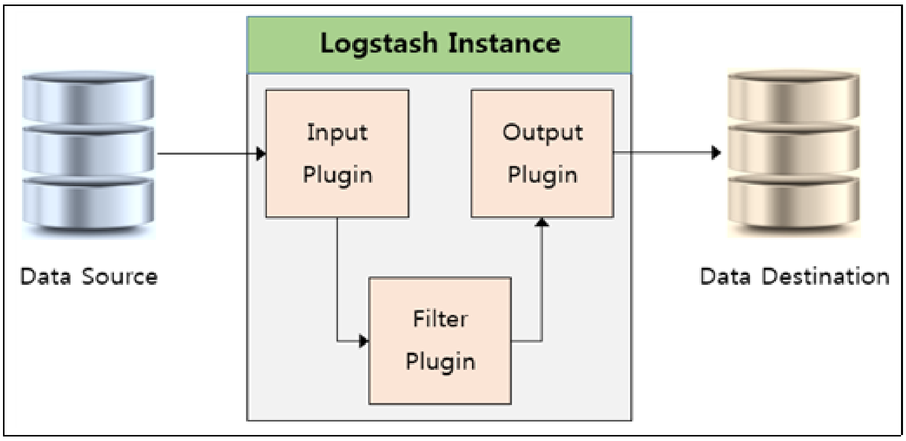


Figure 3. Elasticsearch Cluster Configuration.

Logstash is a plug-in-based event forwarder that has various functions. It can collect data from multiple sources simultaneously and send the data to other places after the conversion, and each event is processed through 3 step pipelines as follows [19]. Firstly, it can process a certain input stream as an input plug-in, it supports various input types, and it can capture events in TCP/UDP socket, HTTP API end point, Elasticsearch, CSV file and all general data sources. Secondly, complex jobs can be performed as it provides a plug-in for various data jobs and applies it conditionally. Thirdly, in case of displaying the data, it supports the extensive range of types.

Although all events are sent to the Elasticsearch, the Logstash can independently store the data through SQL database, CSV file, and data analysis algorithm, and it can also display in the console for the purpose of debugging.

2.2.3. Kibana

Kibana is an expanded type interface, which visualizes web-based dialogue-type data in detail [13,14]. Also, according to an official document of Elasticsearch, the Kibana is an open-source analysis and visualization platform for structured and unstructured data that is designed to interoperate with the Elasticsearch. It can get the data from the Elasticsearch, visualize such data based on the user definition, and provide tools necessary to display in the intuitive dashboard [20].

It offers the visualization of all types of data; provides lines, bars, and distributed graphs which are easier for users to understand by intelligently performing mathematical analysis and conversion on the data; and can creatively visualize the geographical information and location data by utilizing the system data sharing, the flexible interface, and the map service.

2.2.4. FileBeats

Filebeat is a part of Elastic Stack that is known as Beats, and it is one of light-weight data shippers. Beats data Shipper is designed to be installed in the machine that creates the data without effecting the performance of the machine with a single purpose, and it can read text-based log files and send them to Elasticsearch or Logstash [21].

3. Proposed Model

3.1. Model Configuration

The following [Figure 4] is a system configuration diagram that is proposed in this study. Many CentOS 7 version of Linux-based servers runs Apache Web Server, and they record logs related to the user requests. ELK Stack 6 version which is composed of Logstash, Elasticsearch, Kibana is installed in the central server, and the database is installed and configured with MySQL 5.7 version.

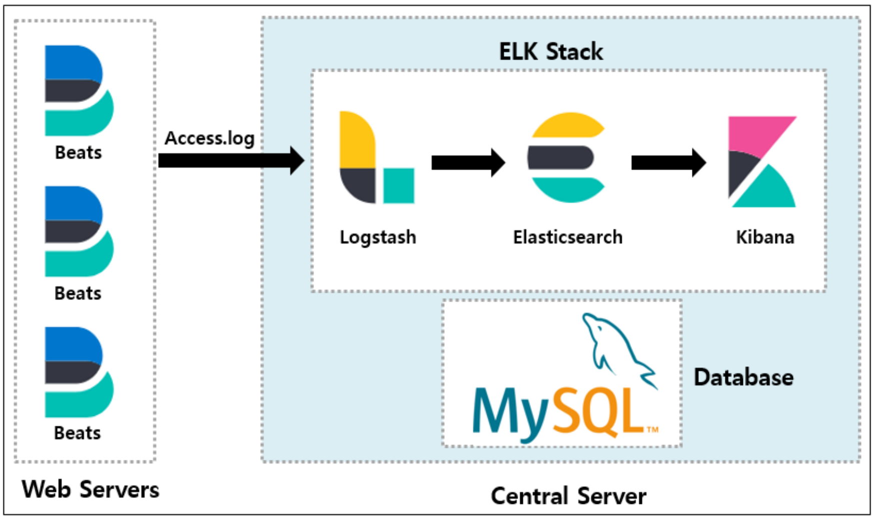


Figure 4. System Configuration Diagram.

3.2. System Flow Diagram

The following [Figure 5] is a system flow diagram that is proposed in the study, which is composed of 4 steps in large. Firstly, it sends the access.log data that is recorded by detecting /var/log/httpd/access.log which stores the access records of each web server through Beats to Logstash of the central server. Secondly, the time information, the service IP information and the host domain information get extracted from the logs that have been received from Beats through the filter in the Logstash. The extracted data is sent to Elasticsearch of the central server. Thirdly, the Elasticsearch extracts the unique value of service IP that corresponds to the host domain by analyzing logs that have been collected. Fourthly, the extracted host domain and Service IP data are compared to the status data that is stored in the DB. Fifthly, the status data and omission detection data are shown by visualizing them based on the user settings.

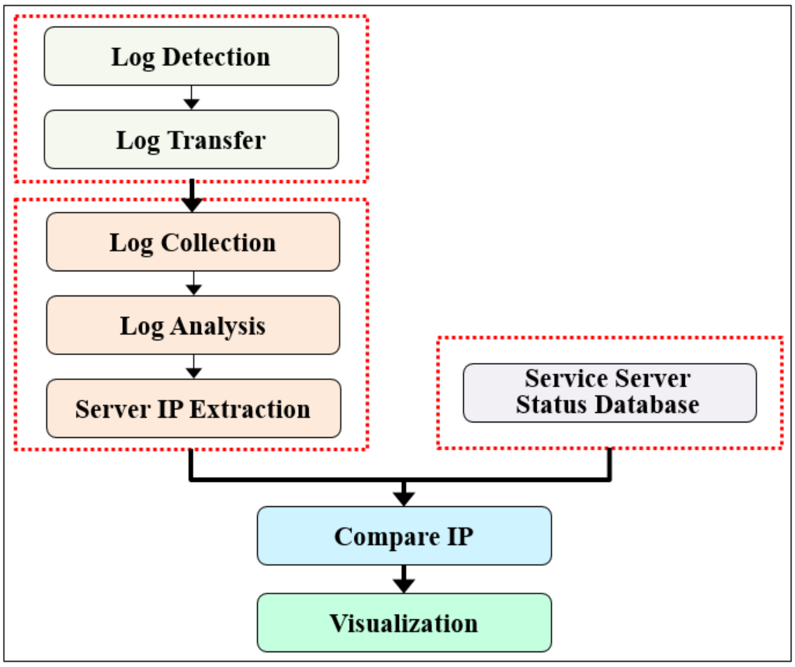
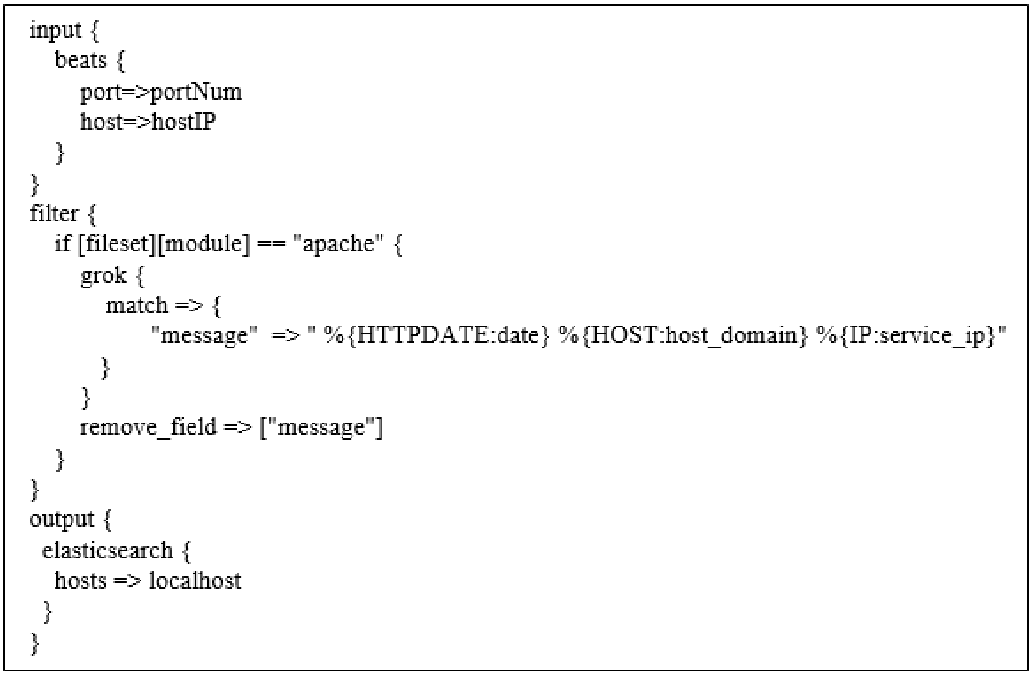


Figure 5. System Flow Diagram

Logstash configuration is composed of input, filter, and output. The input configures the Host IP and the port number to be received from Beats. The filter determines the rule to make a filter on the logs that are received from the Beats.

In this study, if fileset and module are Apache, the HTTPDATE type shall be designated to be date, the HOST type shall be designated to be host_domain, and the IP type shall be designated to be service_ip. Since the content of a message is not used afterward, it shall be deleted with remove_field. The output designates the location that the data to which the filter has been applied would be sent to. In this configuration, it was designated to be localhost where the Elasticsearch is installed. The following [Figure 6] is an algorithm to send access and error logs that have been collected in the Filebeat module by the Logstach pipeline configuration and analyze the syntax.



```
input {
   beats {
      port=>portNum
      host=>hostIP
   }
}
filter {
   if [fileset][module] == "apache" {
      grok {
         match => {
              "message"  => " %{HTTPDATE:date} %{HOST:host_domain} %{IP:service_ip}"
         }
      }
      remove_field => ["message"]
   }
}
output {
  elasticsearch {
   hosts => localhost
  }
}
```

Figure 6. Logstash Pipeline Configuration Algorithm

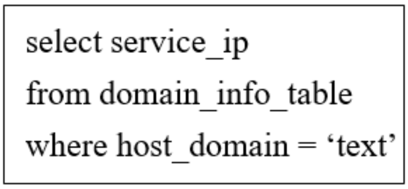
3.3. Database Configuration

The following [Table 1] is a structure of the table that contains IP data which services the host domain. Index is a field that is a unique key value of the host domain and service IP that are registered as the sequence data. host_domain is a name data field of the host domain, and service_ip is an IP address field of the server that executes the host domain service.

Table 1. DB Table Information.

Field	Type	Description
index	int(10)	index
host_domain	varchar2(255)	Host Domain name
service_ip	varchar2(50)	Service Server IP Address

The following [Figure 7] is an algorithm for searching Service IP.



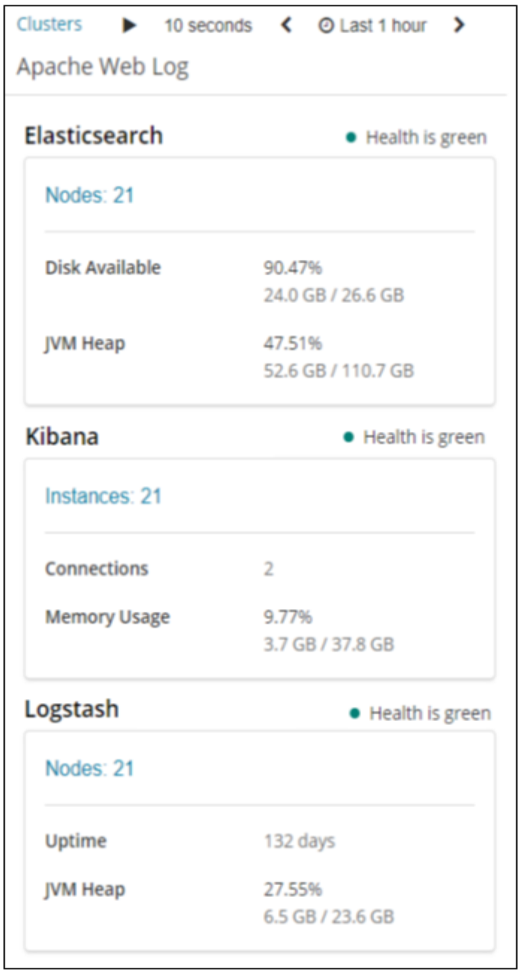
```
select service_ip
from domain_info_table
where host_domain = ‘text’
```

Figure 7. Service Search Algorithm

By using data of the host domain that is used during the log analysis, the Service IP list is searched in the service status table. The data that has been searched is used when finding the service IP that had been omitted from the collection by comparing with the IP that has been acquired through the log analysis.

4. Results and Discussion

The following [Figure 8] is a system monitoring page that can check the overall status of Elasticsearch, Kibana, and Logstash by using Kibana.



Apache Web Log

Elasticsearch ● Health is green

Nodes: 21

Disk Available 90.47%
24.0 GB / 26.6 GB

JVM Heap 47.51%
52.6 GB / 110.7 GB

Kibana ● Health is green

Instances: 21

Connections 2

Memory Usage 9.77%
3.7 GB / 37.8 GB

Logstash ● Health is green

Nodes: 21

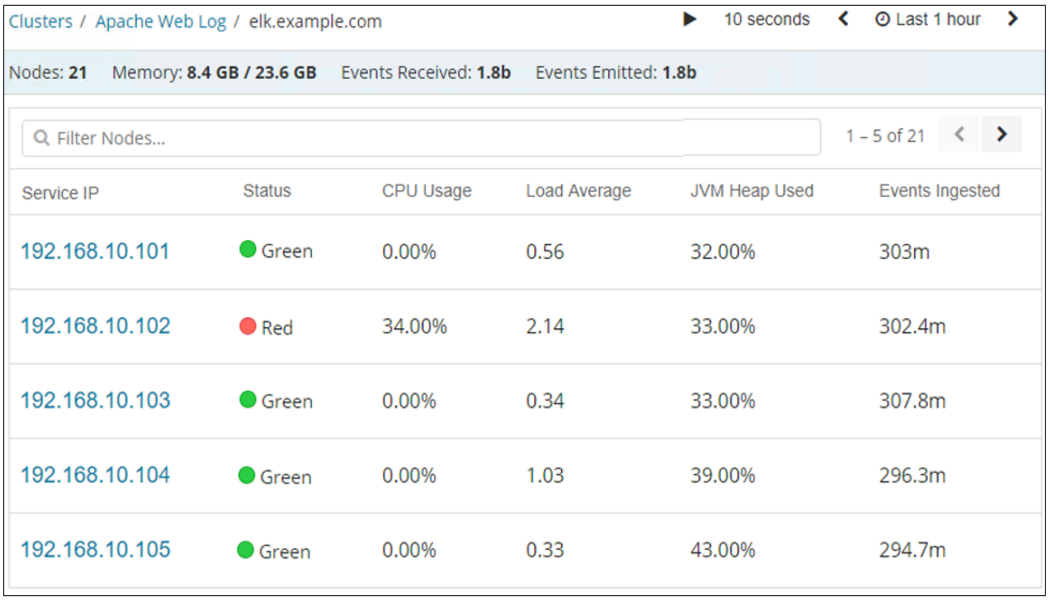
Uptime 132 days

JVM Heap 27.55%
6.5 GB / 23.6 GB

Figure 8. Module monitoring

Health section is composed of green and red, and the status of each group can be checked. As for the monitoring items in the Elasticsearch, the amount of use of the disk is checked in order to see the size to query, and the amount of use of the memory is checked in order to check the Heap section where the instance and array of JVM (Java Virtual Machine) are stored. The monitoring items in the Kibana include the number of connections and the amount of use of memory. The monitoring items in the Logstash include uptime that can check the service initiation time and JVM Heap which is same as the Elasticsearch.

The following [Figure 9] is a program screen that monitors the collection status of each server and system indices at the same time by using Kibana. Service IP status is shown, and the collection status of the related service is shown in Status. If the collection is not taking place, it is shown as a red circle with 'Red' text, and if the collection is taking place, it is shown as a green circle with 'Green' text. Additionally, the amount of use, Load Average, JVM Heap Used, and Events Ingested of CPU of each server can be checked, and the status of each server can be checked at the same time.



Clusters / Apache Web Log / elk.example.com

Nodes: **21** Memory: **8.4 GB / 23.6 GB** Events Received: **1.8b** Events Emitted: **1.8b**

Service IP	Status	CPU Usage	Load Average	JVM Heap Used	Events Ingested
192.168.10.101	● Green	0.00%	0.56	32.00%	303m
192.168.10.102	● Red	34.00%	2.14	33.00%	302.4m
192.168.10.103	● Green	0.00%	0.34	33.00%	307.8m
192.168.10.104	● Green	0.00%	1.03	39.00%	296.3m
192.168.10.105	● Green	0.00%	0.33	43.00%	294.7m

Figure 9. Log collection monitoring system.

5. Conclusion

The number of servers that provide web services is increasing as the number of internet users increases. As the number of servers that provide web services is increased, the number of servers the administrator has to manage is also increased as well. When an issue occurs in the web service, the administrator has to take a look at the logs and find a way to solve the issue, but it takes a lot of time resources while checking each log of many servers by accessing ssh of the servers. In order to solve this problem, a plan to enhance the reliability of the log analysis data through a system that can monitor whether there is any server of which the collection has been omitted by using access log analysis data of Apache web server using ELK is proposed in this study. Through the proposed model, the administrator can reduce the time resources that are consumed for collecting the logs, and it is expected to provide smooth service to users by reducing the time to identify the cause and take necessary measures since integrated logs are checked when the log data needs to be checked due to the occurrence of a failure. As for the future study, the study that can establish the process which can automatically take necessary actions through the analysis of the logs that are recorded during the occurrence of failures shall be continued.

References

[1] Cooper, J., & Brewerton, G. Developing a prototype library WebApp for mobile devices. Loughborough University. 2013.

[2] Alspaugh, S., Chen, B., Lin, J., Ganapathi, A., Hearst, M., & Katz, R. (2014). Analyzing log analysis: An empirical study of user log mining. In 28th Large Installation System Administration Conference (LISA14). 2014 Nov;52-68.

[3] Lee, G., Lin, J., Liu, C., Lorek, A., & Ryaboy, D. The unified logging infrastructure for data

analytics at Twitter. arXiv preprint arXiv:1208.4171. DOI:10.14778/2367502.2367516.

[4] Yu, X., Joshi, P., Xu, J., Jin, G., Zhang, H., & Jiang, G. (2016). Cloudseer: Workflow monitoring of cloud infrastructures via interleaved logs. ACM SIGARCH Computer Architecture News, 44(2), 489-502. DOI:10.1145/2980024.2872407.

[5] Fu, Q., Lou, J. G., Wang, Y., & Li, J. (2009, December). Execution anomaly detection in distributed systems through unstructured log analysis. In 2009 ninth IEEE international conference on data mining, IEEE. 2009 Dec;149-158. DOI:10.1109/icdm.2009.60.

[6] Xu, W., Huang, L., Fox, A., Patterson, D., & Jordan, M. I. (2009, October). Detecting large-scale system problems by mining console logs. In Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles. 2009;117-132. DOI:10.1145/1629575.1629587.

[7] Du, M., Li, F., Zheng, G., & Srikumar, V. (2017, October). Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017 Oct. 1285-1298. DOI:10.1145/3133956.3134015.

[8] Michael, C. C., & Ghosh, A. (2002). Simple, state-based approaches to program-based anomaly detection. ACM Transactions on Information and System Security (TISSEC), 5(3), 203-237. 2002 Aug;5(3)203-237. DOI:10.1145/545186.545187.

[9] Debnath, B., Solaimani, M., Gulzar, M. A. G., Arora, N., Lumezanu, C., Xu, J., et al. LogLens: A real-time log analysis system. In 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS) , IEEE. 2018 Jul;1052-1062. DOI: 10.1109/icdcs.2018.00105

[10] Kuruba, M., Shenava, P., James, J. Real-time DevOps Analytics in Practice. International Workshop on Quantitative Approaches to Software Quality(QuASoQ). 2018 Dec;42-47.

[11] Mavridis, I., & Karatza, H. (2017). Performance evaluation of cloud-based log file analysis with Apache Hadoop and Apache Spark. Journal of Systems and Software, 2017;125:133-151. DOI:10.1016/j.jss.2016.11.037.

[12] Apache Access Log [Internet]. Available from: https://httpd.apache.org/docs/2.2/en/logs.html.

[13] Moriyama, K., Nakatani, T., Yasu, Y., Ohshita H., Seya, T. Development of Status Analysis System Based on ELK Stack at J-PARC MLF. 16th International Conference on Accelerator and Large Experimental Control Systems. 2018. DOI: 10.18429/JACoW-ICALEPCS2017-THPHA033.

[14] Mikula A., D Adamová, M., Adam, J., Chudoba, J., Švec, J. Grid Site Monitoring and Log Processing using ELK. Institute of Physics of Czech Academy of Sciences. 2016;54-61.

[15] Elasticsearch, Kibana [Internet]. Available from: https://www.elastic.co/elastic-stack.

[16] Appleyard, R., & Adams, J. Using the ELK Stack for CASTOR Application Logging at RAL. In International Symposium on Grids and Clouds 2015 SISSA Medialab. 2016 Mar;239:1-15.

[17] Procopiuc, O., Agarwal, P. K., Arge, L., & Vitter, J. S. Bkd-tree: A dynamic scalable kd-tree. In International Symposium on Spatial and Temporal Databases Springer, Berlin, Heidelberg. 2003 Jul;46-65.

[18] Gormley, C., Tong, Z. Elasticsearch: the definitive guide: a distributed real-time search and analytics engine. Sebastopol: O’Reilly Media, Inc. 2015.

[19] Bajer, M. Building an IoT data hub with Elasticsearch, Logstash and Kibana. In 2017 5th International Conference on Future Internet of Things and Cloud Workshops. 2017Aug;63-68. DOI:10.1109/FiCloudW.2017.101.

[20] Diaz. A. Development of a monitoring system for the DQMGUI in ElasticSearch and Kibana. European Organization for Nuclear Research (No. CERN-STUDENTS-Note-2016-210). 2016 Sep; 1-5.

[21] Hamilton, J., Gonzalez Berges, M., Tournier, J. C., Schofield, B. SCADA Statistics monitoring using the elastic stack (Elasticsearch, Logstash, Kibana). 16th International Conference on Accelerator and Large Experimental Control Systems. 2017;451-455. DOI:10.18429/JACoW-ICALEPCS2017-TUPHA034.