Oleksandr MALAKHOVSKYI[1], Ihor BARAN[2]

Supervisor: Mikołaj KARPIŃSKI [3]

SCENTRALIZOWANE LOGOWANIE W ŚRODOWISKU KONTENERA

Streszczenie: W artykule omówiono podejście do scentralizowanego zbioru dzienników i analizy we wszystkich środowiskach kontenerowych. Oznacza to, że do rejestrowania aplikacji i zarządzania logistyką aplikacji kontenerowych będzie używany tylko jeden system logowania.

Słowa kluczowe: log, gromadzenie, elasticsearch, logstash, kibana, ELK Stack, kubernetes, Openshift, kontener

CENTRALIZED LOGGING IN CONTAINER ENVIRONMENT

Summary: This paper discussed an approach to a centralized log collection and analyzation in all container environment. That means only one logging system will be used for application logging and for management of containerized applications platform logging.

Keywords: log, collecting, elasticsearch, logstash, kibana, ELK Stack, kubernetes, Openshift, container

1. Introduction

Logs are a critical part of any software development process, they give a deep insight about an application and application lifecycle, what happens in a system and what exactly caused the error when something incorrect happens. A centralized log management and analysis strategy are mission critical for each system, enabling organizations to understand the complex relationship between operational, security,

[1] M.Sc. Student, Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, department of Computer Science, oleksandr_malakhovskii@tstu.edu.ua

[2] Ph.D., Assoc. Prof., Faculty of Computer Information Systems and Software Engineering, department of Computer Science, a head of department, ihor.remm@gmail.com

[3] Prof. D.Sc., University of Bielsko-Biala, Faculty of Mechanical Engineering and Computer Science, department of Computer Science and Automatics, a head of department, mkarpinski@ath.bielsko.pl

and application events. Building enterprise level application, a system goes to multiple hosts and many servers, managing the logs across that system can be complicated and slow. Furthermore debugging the error in the application across thousands of log files on hundreds of servers can be very complicated and sometimes time-consuming. A common approach to this problem is building a centralized logging system which can collect and aggregate different logs in one location and system.

2. Collecting, analyzing Logs from containerized applications and container management system

The first stage of solving the problem of centralized logging in a container environment is a logs collection from different sources. Containers and container management platforms like Kubernetes or Openshift produce logs in different ways, for example through syslogs and other logs directly in files. Log files can also be different formats, like JSON or plain text.
The second stage is processing and filtering collected data. Log files can consist of thousands line of text and numbers, furthermore many fields are not important for logging system. Logs in text format are unindexed and sometimes don't have any structure. In contradistinction to text logs, logs in JSON format have more features. For example, all fields are already in "key=value" format, that means we can quickly find and select the requested field. Almost every system and programming language support JSON natively. Usually, a log entry includes such information as: the date and time the event occurred, the container the event occurred on. There are many ways available to transport log data. One way is directly plug input sources and framework can start collecting logs and another way is to send log data via REST API [1], application code is written to log directly to these sources it reduces latency and improves reliability.

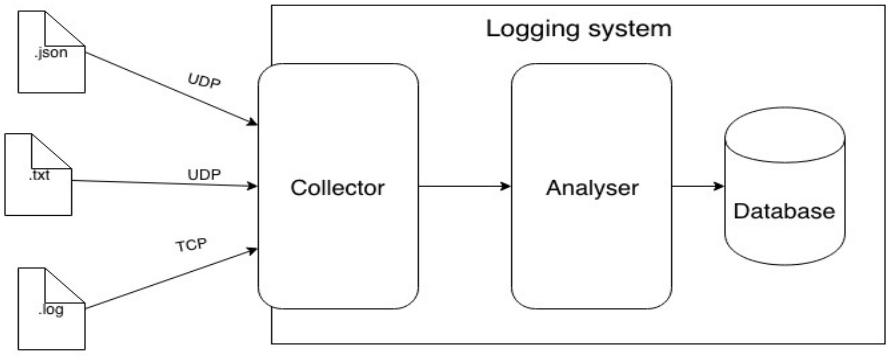

Figure 1 – Transporting and analysing logs

A common approach in container world is a syslog driver for transporting. Syslog is a standard driver for message logging in Docker[2], that means a container can send logs per TCP or UDP independently of any frameworks or log rotation tools.
Logging system should filter and parse a data, after that find and save a required data. Log analytics occurs by organizing data via processing text data, tagging and storing

as indexed text. There are many different search and analytics engines. One of the most popular is elasticsearch. Elasticsearch is a highly scalable free and open-source full-text search and analytics engine. It can store, search, and analyze big volumes of data quickly and it works really good with real-time data[3]. Elasticsearch can be used to search all types of documents and files. It provides a fast and scalable search, has near real-time search, and supports multi tenancy and high-availability.
Unfortunately, Elasticsearch is only searching and indexing engine, it can't collect and filter data from different sources. Elasticsearch is developed together with a data-collection and log-parsing engine called Logstash, and an analytics and visualization tool called Kibana. The three products are designed for use as an integrated solution, referred to as the "Elastic Stack" (formerly the "ELK stack"). In ELK stack Logstash is an entry point for all incoming data. Logstash is an open source, server-side data processing tool that ingests data from a different sources simultaneously, transforms it, filter it and then sends it to Elasticsearch.
ELK stack also has a tool called Beats. Beats is the tools for single-purpose data shippers. They like lightweight agents and send data from hundreds or thousands of sources to Logstash or Elasticsearch directly [4]. Beats can send many different types of data. For example, they can send a physical server metrics, network traffic or files to elasticsearch. But Beats are only data senders, they can't filter or process input data.
Generally, Logstash is a default input and filtering tool. The Logstash input data processing process has always three stages: inputs, filters, and outputs. Each Input generates a single event, then filter modify it, and outputs ship them elsewhere, normally to elasticsearch. Inputs and outputs support different codecs that enable the possibility to encode or decode the selected data as it enters or exits the process without having to use a separate filter or any other tool. Logstash filters are intermediary processing tools in the ELK Stack. They can combine filters with conditionals to perform an action on an event if it meets specials criteria. The last tool is Kibana. Basically, Kibana is an open source data visualization REST tool for Elasticsearch API. It provides visualization capabilities on top of the already indexed content in Elasticsearch.

3. Log senders and log processing

The first and the most important log senders are docker container with installed applications. The can send thousand logs in the same time, but docker provides very flexible logging configuration. The main log format for containers will be JSON format and a logging driver Syslog. Containers will send log data directly to Logstash per UDP protocol. Note, that can UDP uses a simple connectionless communication model with a minimum of protocol mechanism and It has no handshaking dialogues. That means sending will be fast, but UDP does not guarantee a correct data transfer. In this case, it's not a critical problem, because all containers will run on Openshift Container Platform. OpenShift is a container-based software deployment and management product from Red Hat. It`s open-source project based on Kubernetes from Google. OpenShift Container Platform uses a software-defined networking (SDN) to provide a unified and flexible cluster network that enables communication

between containers across the whole OpenShift Container Platform cluster[5]. This container network is established and maintained by the OpenShift SDN, which configures an overlay network through Open vSwitch (OVS) mechanism. Openshift provides a stable networking inside a cluster, which make a transferring per UDP stable and secure. Each Openshift node has an own SDN Router with NAT Networking. Compute nodes do not have directly network access, communication with the external world can be only through the main OpenShift router on the master node.

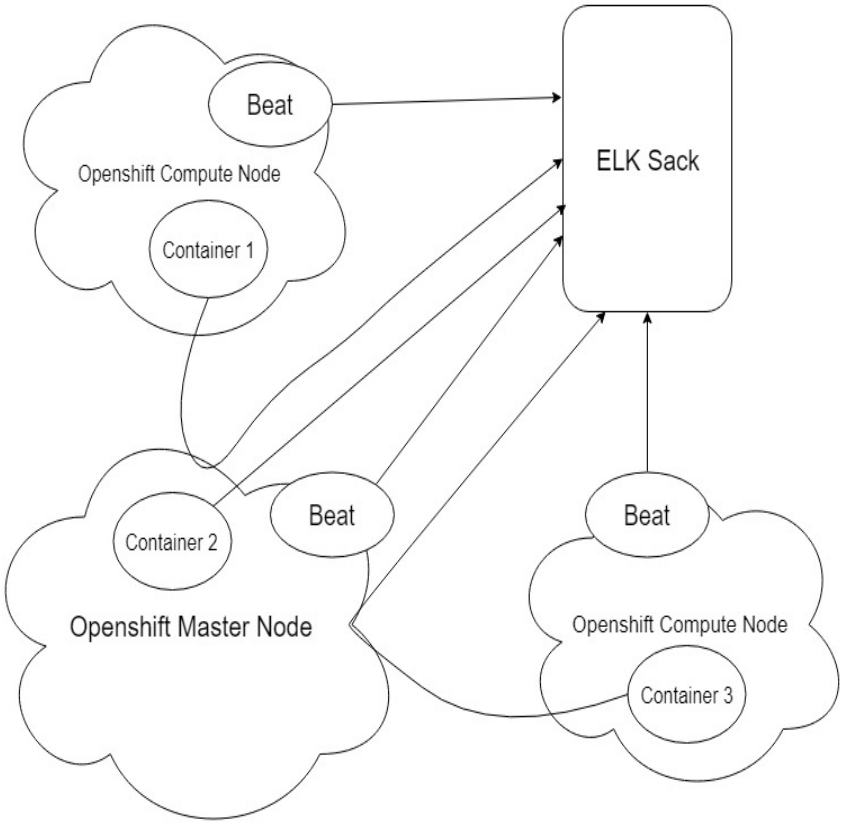

Figure 2 - Log sending

The second sender is a container management platform Openshift. Unfortunately, Openshift cannot send logs directly per UDP to ELK Stack. It uses a standard Linux journal for logging[6]. That means Beats agent should be installed on each Openshift node.

The next step is text data processing. First of all Logstash check a log type, because Openshift sends logs in text format, but containers sending them in JSON format. Then Logstash will find predefined fields in input file, select them and send to Elasticsearch. Logstash can also send whole file content but in most cases it's not

required.Note that processing JSON files are a little bit faster than processing text files. Usually Logstash search for a key in JSON file, but text file should be read word by word. In the next step Elasticsearch will index data and Kibana visualize that data in a browser.

4. Conclusions

This paper proposes an approach to centralized monitoring in a container environment. Also, we show how to work with different type of log data, how to send them and work with them. As you can see different log type and log file content is not a big problem for Elastic Stack. Via Logstash we can clean or format our log data and then send them to Elasticsearch for indexing. Furthermore, we have a lot of possibilities in data analysis and visualization in Kibana. We can send log data to Logstash in many different ways, such as directly per network or with different senders like Beats.
In future work, we plan to deploy that system and test it in real-world cases. We also plan experiments with different docker management systems and we will try to use ELK Stack also as a monitoring solution.

REFERENCES

1. MARK MESSE: REST API Design Rulebook, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2012. 47–52.
2. Configure logging driver | Docker docs - Docker official Documentation: *https://docs.docker.com/config/containers/logging/configure/* 28.10.2018
3. Elasticsearch: RESTful, Distributed Search & Analytics - Elasticsearch Documentation: *https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html* 29.10.2018
4. Getting started with Beats - Elasticsearch documentation: *https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html* 26.10.2018
5. GRAHAM DUMPLETON.: Deploying to OpenShift, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472 2017, 93-101
6. Aggregating Container Logs - Openshift Installation and Configuration docs: *https://docs.openshift.com/container-platform/3.3/install_config/aggregate_logging.html*