

LOGGERVICK (USAGE GUIDELINES)



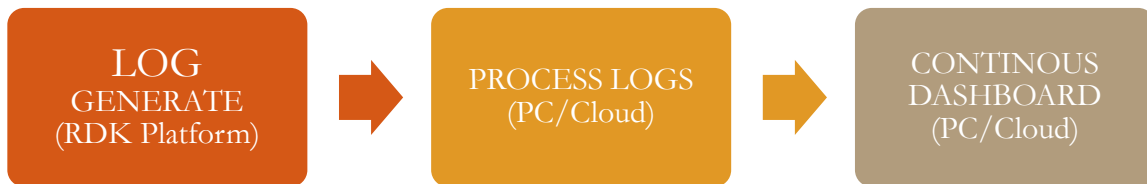
Team name: Atlas
Member's name: Mayank,
Muthu, Sembulingam &
Thiru
Date: September 16, 2021

Table of Contents

INTRODUCTION	2
LOG GENERATION	2
LOG EMULATOR	3
OVERALL SETUP	3
ELK SETUP	4
HOW TO IMPORT DASHBOARD IN KIBANA	4
DASHBOARD OUTPUT	0
ADVANTAGES OF THE APPROACH	0
IMPROVEMENTS	0
BIBLIOGRAPHY	I

INTRODUCTION

LoggerVick is a profiling tool, that uses ELK stack along with syslog traces to analyze and perform the profiling in the external machine. External machine can be a different PC or Cloud, where ELK can be installed. Below is the sequence of the execution to get the profiling information in a Dashboard.



LOG GENERATION

Idea is to use the RDK Platform only to generate the traces, and thereby the profiling is done externally. Approach reduces un-wanted addition in the code, provides serialized history about the logs generated with their information. Specific format is expected by the profiler's waiting outside the system. Below is expected syntax of the logs,

<TimeStamp> <DeviceName> <Process> <ThreadID> <FileName: Line> <Function Name> <Enter | Exit>

2021 Sep 09 07:26:28.432511 raspberrypi-rdk-mc WPEFramework[980]: [1025] INFO
[Bluetooth.cpp:1183] getDiscoveredDevicesWrapper: Enter

2021 Sep 09 07:26:28.432511 raspberrypi-rdk-mc WPEFramework[980]: [1025] INFO
[Bluetooth.cpp:1183] getDiscoveredDevicesWrapper: Exit

As seen above with an example, Syntax was defined considering minor change, precisely only the addition of “Enter” and “Exit” as the function entry and exit respectively. Expectation for analyzes is to add these traces to the functions that required profiling. RDK Platform is expected to stream the traces to the PC | Cloud, that is hosting the ELK. For the demo purpose the UDP streaming is enabled in the RDK using the “loggervickstreamer.sh”. The shell file takes IP, Port and filename to be streamed as arguments. Once started the “loggervickstreamer” continuously sends logs to the ELK hosted machine. Approach was to use RDK platform only to bookmark the events and send to the external machine.

LOG EMULATOR

As part of process of verifying with the traces, we have developed a “log_emulator.sh” script. This was important for the ELK team to manage the development exclusively without the RPi or the RDK Platform. We have managed to execute this script in the RPi also. Script generates the log in the specific format.

USAGE : `log_emulator.sh >> /opt/logs/testlog.sh`

This was usual execution sequence before logs were enabled in the RPi modules

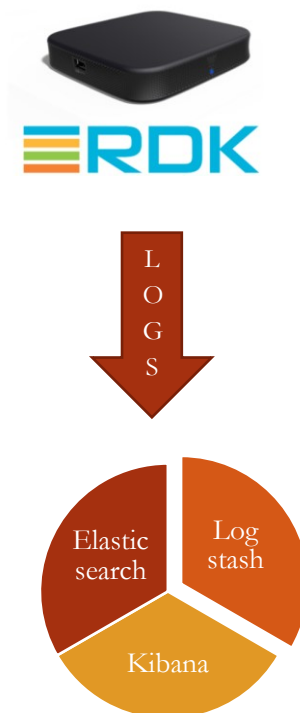
USAGE:

`touch testlog.sh`

`loggervickstreamer.sh 192.168.x.x 5044 testlog.sh &`

`log_emulator.sh >> /opt/logs/testlog.sh`

OVERALL SETUP



ELK SETUP

ELK setup shall manage profiling by calculating the time taken between “Enter” and “Exit” bookmark logs, using the “Elapsed filter plugin”. ELK requires setting up Elasticsearch, Logstash and Kibana setup either in same host or different host or in cloud. Logstash is waiting to read the data available in a specific port (5044) as defined in the “`rdkloganalyzer.conf`”. ELK as well established in available in all OS environments. Recommend adding bin path to the Environment variables for easy execution.

Logstash shall filter valid and start different timer for each function and calculate elapsed time for each function. This is usually a work performed in the RDK platform, we have managed handle it is using an Elapsed filter Plugin in Logstash. So Logstash default installation requires 2 additions

- Elapsed filter plugin (Link in Bibliography)
- Enable `config.support_escapes : true` in `logstash.v.x.x.x/config/logstash.yml`

After analyzing Logstash adds logs with information and time details (profiling) into the Elasticsearch database with the timestamp with hostname as “Index pattern”. All 3 Components are to be started before streaming the traces using CMD in Windows.

- `elasticsearch`
- `Logstash -f rdkloganalyzer.conf --path.config data`
- `kibana`

Elasticsearch is a database component managing all the logs in the library.

Launch Kibana and import provided json file to add the visual components for the dashboard. Below is the snapshot of the database. To view the data, index pattern used in the Logstash configuration file and key selected in the Kibana shall be same.

ELK Parts

- `rdkloganalyzer.conf`
- `exportrdk.ndjson`

HOW TO IMPORT DASHBOARD IN KIBANA

Launch Kibana -> Stack Management -> Saved Objects

Select Import – Provide `exportrdk.ndjson` as input.

Dashboard : `rdk_dash1`

Index Pattern : `test*`

DASHBOARD OUTPUT

Below is the snapshot of the Dashboard from the Kibana



ADVANTAGES OF THE APPROACH

ELK can be executed either in a PC or CLOUD, thereby providing history and flexibility.

Kibana Dashboard is managed using the timestamp thereby the analyzer can view the past data about profile details compare across different timelines.

Kibana Dashboard also supports profiling across different host for ex : Different devices, if the hostname is different (rdk-generic-wpe-mc can be differentiated with rdk-generic-wpe-hybrid).

Moving the profiling mathematics outside the system means, updates and improvements on analyzes can be done without adding changes in the RDK platform.

Lot of storage and logs across timelines can be maintained in cloud.

Conceptually the Kibana, elasticsearch and Logstash can be separated and hosted in different places.

Kibana provides AI based approach, can extended to identify pattern in the traces, alert in drastic change in metrics.

IMPROVEMENTS

Logs are analyzed with milliseconds even though generation is in microseconds by syslogd.

Elapsed time calculation is also ceiled in milliseconds.

Update of ELK in future should improve above points.

BIBLIOGRAPHY

Details of the installation, useful links and check-ins captured below,

On the References tab, in the Citations & Bibliography group, click Insert Citation for the option to add sources and then place citations in the document.

- [Installing the Elastic Stack | Installation and Upgrade Guide \[7.14\] | Elastic](#)
- [Elapsed filter plugin | Logstash Reference \[7.14\] | Elastic](#)