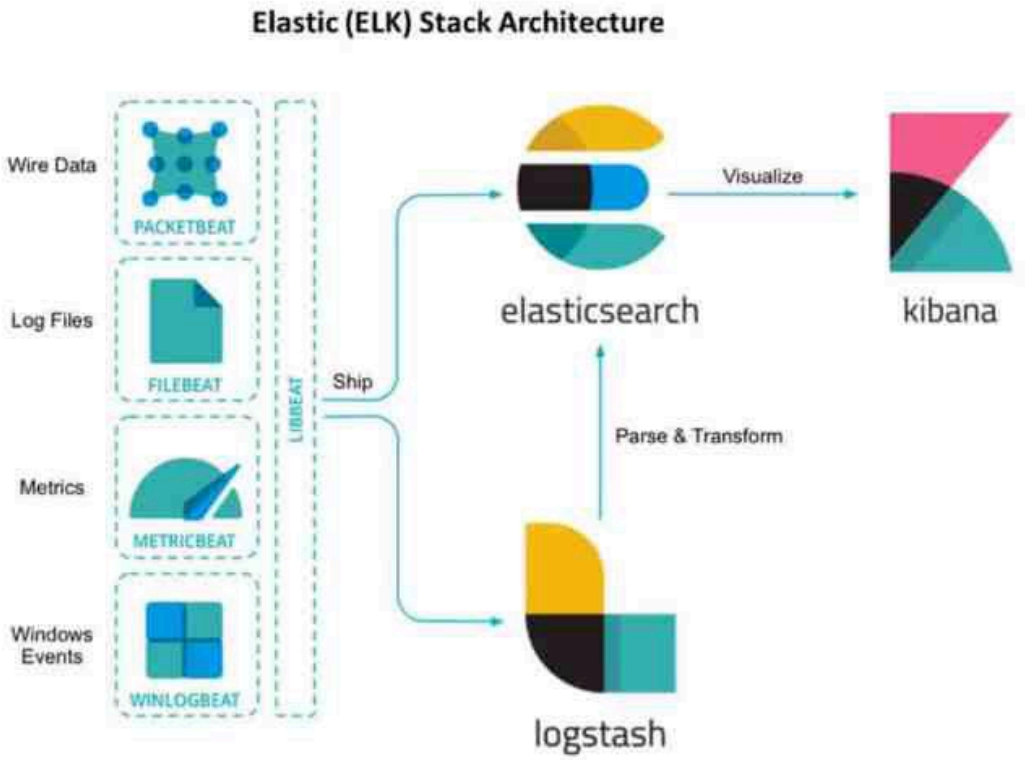
Logstash and Monitoring Data

Logstash and Monitoring Data



Prepared By:

Md Tariqulhasan Fazle Rabbi, TSE, Mulytic Labs GMBH

Sultan Ibrahim Khalil, TSE, Mulytic Labs GMBH

INTRODUCTION

When a system administrator, or even a curious application developer, there is a possible chance that s/he may regularly dig into logs to find precious information among the logs. Sometimes s/he may want to monitor SSH trespass on VMs. Sometimes, s/he might want to see what errors were raised by the application servers on a certain day, on a very specific hour. Or might want to have some insights about who stopped systemd service on one of the VMs. In this document, we are about to set up a complete log monitoring service using the ELK stack (ElasticSearch, Logstash and Kibana) and Rsyslog as a powerful syslog server.

Logstash is an open source data collection engine with real-time pipelining capabilities. Logstash can dynamically collect data from disparate sources and normalize the data into destinations of your choice. Cleanse and democratize all your data for diverse advanced downstream analytics and visualization use cases.

While Logstash originally designed for innovation in log collection, its capabilities extend well beyond that use case. Any type of event can be enriched and transformed with a broad array of input, filter, and output plugins, with many native codecs further simplifying the ingestion process. Logstash accelerates your insights by harnessing a greater volume and variety of data.

IMPORTANCE

Monitoring Linux logs is not so simple and every DevOps engineer should know how to do it. Here’s some reason :

Real-time visual feedback

Probably one of the key aspects of log monitoring, one can build meaningful visualizations (such as datatables, pies, graphs or aggregated bar charts) to give some meaning to logs.

Able to aggregate information

Sometimes raw information is not enough, users may want to join it with other logs or to compare it with other logs to identify a trend. A visualization platform with expression handling lets one perform that.

Able to make a quick filter

if you are only interested in SSH logs, you can build a targeted dashboard for it.

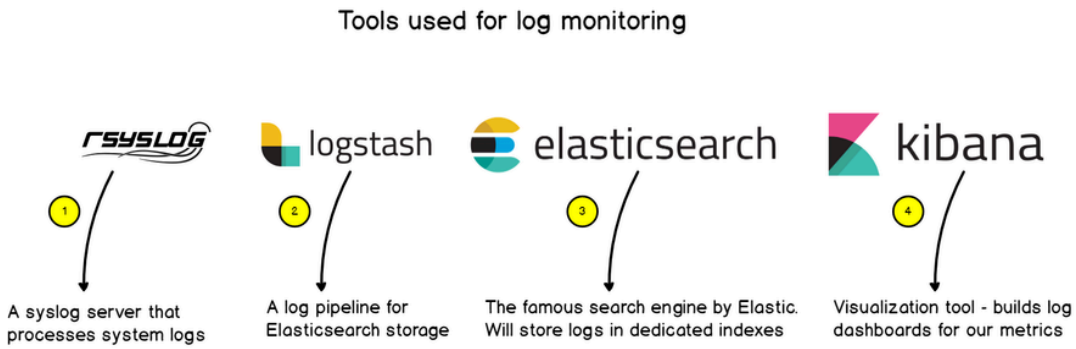
Logs are navigable

Everyone knows the pain of tailing and greping logs files endlessly. It would rather have a platform for it.

BENEFITS

1. How logs are handled on a Linux system Ubuntu and what rsyslog is.
2. How to install the ELK stack (ElasticSearch *.*, Logstash and Kibana) and what those tools will be used for.
3. How to configure rsyslog to forward logs to Logstash
4. How to configure Logstash for log ingestion and ElasticSearch storage.
5. How to play with Kibana to build our final visualization dashboard.

MATERIALS



1. Rsyslog
2. Elasticsearch
3. Kibana
4. Logstash
5. JVM

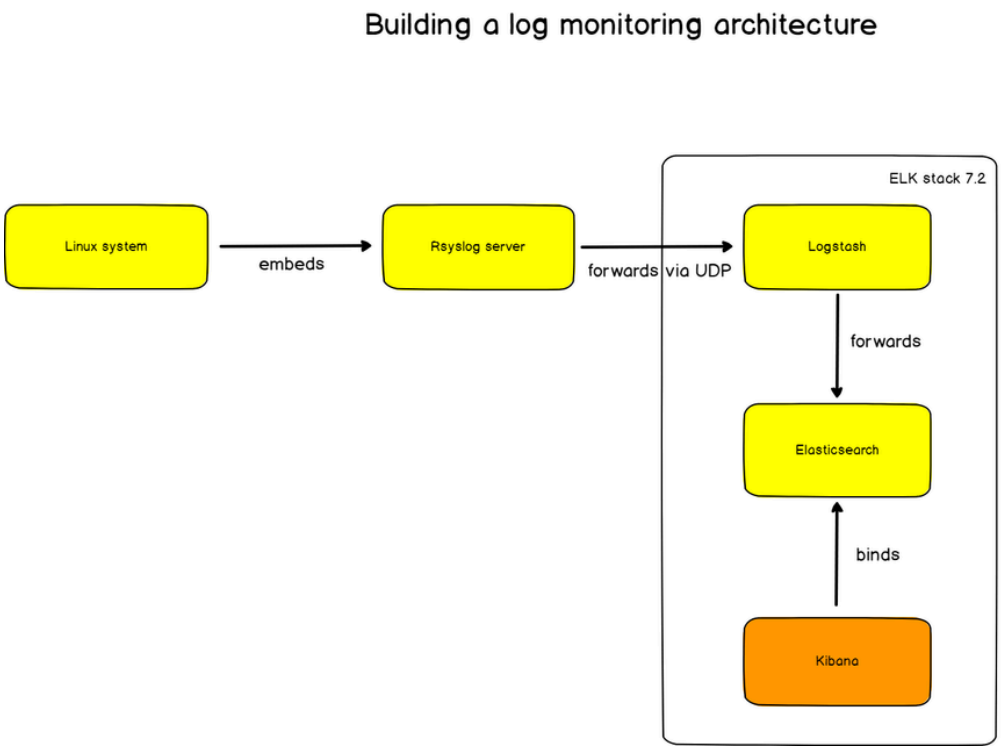
ARCHITECTURE

Now it is time to dig into the architecture of the system.

Key concepts of Linux logging

Historically, Linux logging starts with **syslog.** Syslog is a **protocol** developed in 1980 which aims at standardizing the way logs are formatted, not only for Linux, but for any system exchanging logs. From there, syslog servers were developed and were embedded with the capability of handling syslog messages. They rapidly evolved to functionalities such as **filtering**, having **content routing abilities**, or probably one of the key features of such servers : **storing logs** and rotating them.

Rsyslog was developed keeping this key functionality in mind : **having a modular and customizable way to handle logs.** The modularity would be handled with modules and the customization with log templates. In a way, rsyslog can ingest logs from many different sources and it can forward them to an even wider set of destinations. This is what we are going to use in our tutorial.



INSTALLATION PROCEDURE

Rsyslog

Rsyslog is an open-source software utility used on UNIX and Unix-like computer systems for forwarding log messages in an IP network in our case to Logstash. Installation procedure,

```
sudo add-apt-repository ppa:adiscon/v8-devel
sudo apt-get update
sudo apt-get install rsyslog
```

Elasticsearch

Elasticsearch is a search engine based on the Lucene library. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents.

To install it, run the following command:

```
sudo apt-get update && sudo apt-get install elasticsearch
```

To run logstash

Check the current status

```
$ sudo systemctl status elasticsearch.service
```

Then

```
$ sudo systemctl start elasticsearch.service
```

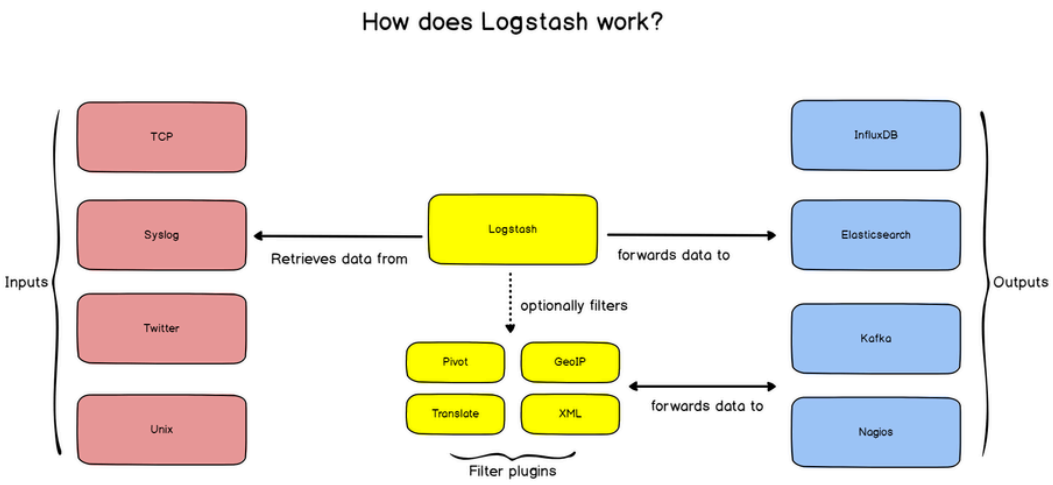
Kibana

Kibana is a proprietary data visualization dashboard software for Elasticsearch and we will use it to monitor our final logs. Wait a min, here's the command to install Kibana:

```
sudo apt-get update && sudo apt-get install kibana
```

Logstash

Logstash is a light-weight, open-source, server-side data processing pipeline that allows us to collect data from a variety of sources, transform it on the fly, and send it to a desired destination in our case the destination is Elasticsearch.



If you added Elastic packages previously, installing Logstash is as simple as executing:

```
sudo apt-get install logstash
```

To run logstash

Check the current status

```
$ sudo systemctl status logstash.service
```

Then

```
$ sudo systemctl start logstash.service
```

JVM

Before installing the ELK stack, you need to install Java on your computer.

To do so, run the following command:

```
$ sudo apt-get install default-jre
```

ROUTING LINUX LOGS TO ELASTICSEARCH

As a reminder, we are routing logs from rsyslog to Logstash and those logs will be transferred to ElasticSearch pretty much automatically.

Routing from Logstash to ElasticSearch

Before routing logs from rsyslog to Logstash, it is very important that we set up log forwarding between Logstash and ElasticSearch.

To do so, we are going to create a configuration file for Logstash and tell it exactly what to do. To create Logstash configuration files, head over to /etc/logstash/conf.d and create a logstash.conf file using nano or what you prefer. Here we are practically implementing the three famous tasks of Logstash Input, Filter and Output.

Inside, append the following content:

```
input {
  udp {
    host => "127.0.0.1"
    port => 10514
    codec => "json"
    type => "rsyslog"
  }
}


# The Filter pipeline stays empty here, no formatting is done.
filter { }

# Every single log will be forwarded to ElasticSearch. If you are using another port, you should specify it here.
output {
  if [type] == "rsyslog" {
    elasticsearch {
      hosts => [ "127.0.0.1:9200" ]
    }
  }
}
```

Note : for this document, we are using the UDP input for Logstash, but if you are looking for a more reliable way to transfer your logs, you should probably use the TCP input. The format is pretty much the same, just change the UDP line to TCP.

Restart your Logstash service.

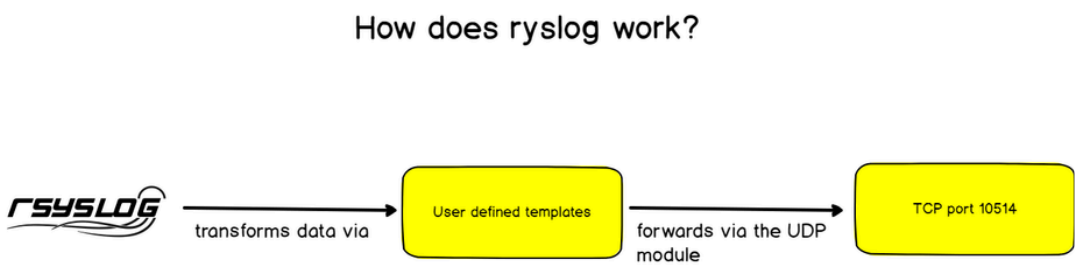
```
$sudo systemctl restart logstash
```

To verify that everything is running correctly, issue the following command:

```
$ netstat -na | grep 10514
udp        0      0 127.0.0.1:10514         0.0.0.0:*
```

Here we are using port 10514 for Logstash i.e Logstash is now listening on port 10514.

Routing from rsyslog to Logstash



As described before, rsyslog has a set of different modules that allow it to transfer incoming logs to a wide set of destinations. Rsyslog has the capacity to transform logs using templates. This is exactly what we are looking for as ElasticSearch expects JSON as an input, and not syslog RFC 5424 strings. In order to forward logs in rsyslog, head over to /etc/rsyslog.d and create a new file named 70-output.conf

Inside your file, write the following content:

```
# This line sends all lines to defined IP address at port 10514
# using the json-template format.
*.*                     @127.0.0.1:10514;json-template
```

Now that you have log forwarding, create a 01-json-template.conf file in the same folder, and paste the following content:

```
template(name="json-template"
  type="list") {
    constant(value="{")
      constant(value="\"@timestamp\":\"")     property(name="timereported" dateFormat="rfc3339")
      constant(value="\",\"@version\":\"1")
      constant(value="\",\"message\":\"")     property(name="msg" format="json")
      constant(value="\",\"sysloghost\":\"")  property(name="hostname")
      constant(value="\",\"severity\":\"")    property(name="syslogseverity-text")
      constant(value="\",\"facility\":\"")    property(name="syslogfacility-text")
      constant(value="\",\"programname\":\"") property(name="programname")
      constant(value="\",\"procid\":\"")      property(name="procid")
    constant(value="\"}\n")
}
```

As you probably guessed, for every incoming message, rsyslog will interpolate log properties into a JSON formatted message, and forward it to Logstash, listening on port 10514.

Restart your **rsyslog service**, and verify that logs are correctly forwarded to ElasticSearch.

Note : logs will be forwarded in an index called logstash-*.

```
$ sudo systemctl restart rsyslog
$ curl -XGET 'http://localhost:9200/logstash-*/_search?q=*&pretty'
{
  "took" : 21,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "logstash-2021.09.12-000001",
        "_type" : "_doc",
        "_id" : "XbDw4nsBGSy6uAhu6GGE",
        "_score" : 1.0,
        "_source" : {
          "message" : " Enter a password for Secure Boot. It will be asked again after a reboot.",
          "severity" : "info",
          "facility" : "daemon",
          "procid" : "6164",
          "programname" : "vboxdrv.sh",
          "type" : "rsyslog",
          "@timestamp" : "2021-09-14T06:14:03.523Z",
          "host" : "127.0.0.1",
          "sysloghost" : "tariq",
```

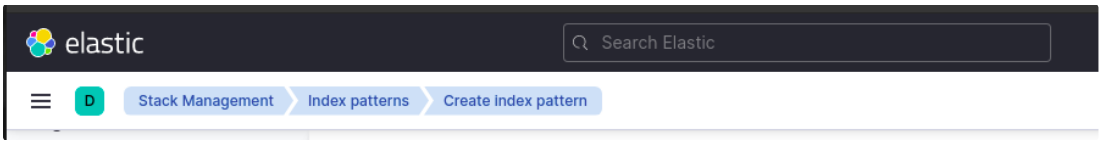
```
      "@version" : "1"
    }
   }
  ]
 }
}
```

Awesome! We now have rsyslog logs directly stored in ElasticSearch.

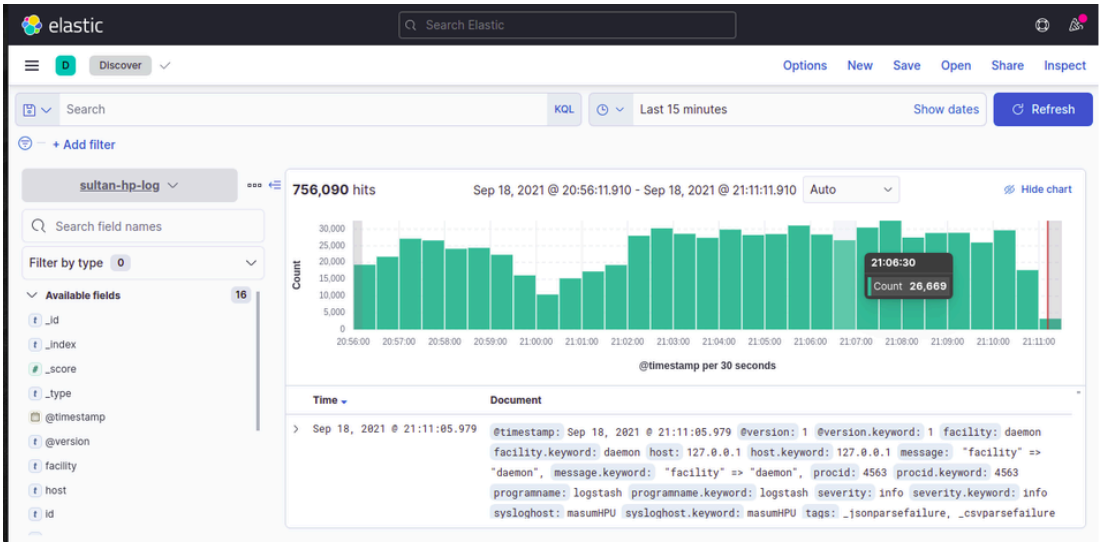
DATA

Now as we've already configured Elasticsearch, Kibana and routed out data to ElasticSearch via Logstash, now we can view/analyze our data in kibana.

At first we have to create an index pattern in KIbana for for data, to do that open kibana in your local machine using http://localhost:5601/ then go the below directory to create an index pattern

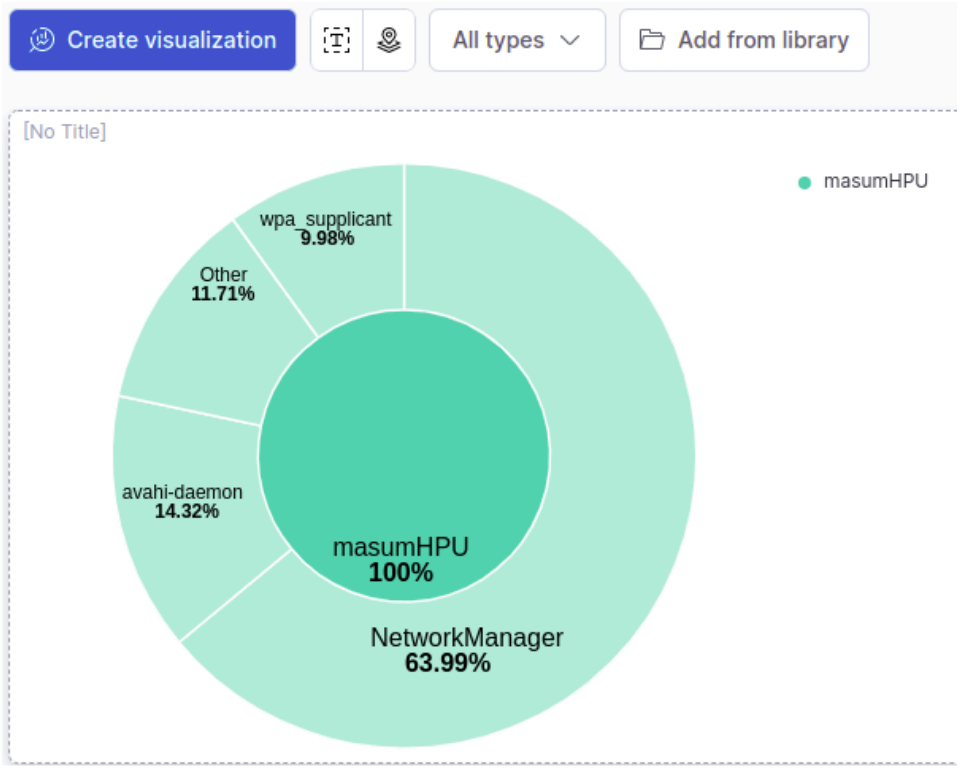


After creating the index patten we can see our dashboard from Discover



in Kibana, just the same way as we monitor our data here in Mulytic. As the rsyslog gets the data from the syslog of the Linux systems, so here we can only see the fields that are passed by the rsyslog.

Along with the Discover we can also create different **Dashboard** with the data, a simple representation is given below,



CONCLUSION

In this document, we tried to give a better understanding of how anyone can monitor an entire logging infrastructure easily with Rsyslog and the ELK stack.

We are also planning to do the same things using docker in Future. Until then, have fun.

REFERENCES

1. *Logstash: Collect, Parse, Transform logs*. Elastic. (n.d.). Retrieved September 12, 2021, from https://www.elastic.co/logstash/.
2. *Getting Started with Logstash | Logstash Reference [master]. (n.d.). Elastic. Retrieved September 12, 2021, from https://www.elastic.co/guide/en/logstash/master/getting-started-with-logstash.html*
3. *Schkn, S. (2019, August 21). Monitoring Linux Logs with Kibana and Rsyslog. Devconnected. https://devconnected.com/monitoring-linux-logs-with-kibana-and-rsyslog/*