Processing millions of logs with Logstash

and integrating with Elasticsearch and Cassandra

Valentin Fischer, SIG-NOC 2015

November 9, 2015

About me

My name is **Valentin Fischer** and I work for the **University of Vienna**.

I'm a **monitoring guy and sys admin**, working daily with systems like Logstash, Icinga, Elasticsearch, Cassandra.

Scope of this presentation

1. This presentation is meant as a **overview** of our current setup.

2. Describing possible **integration of logstash in ”big data”** environments.

Logstash: Why?

Because we needed something to cover the logging part of the monitoring.

Logs are hard and ungly to work with.

This seems like a good solution to manage logs in "modern" way.

Logstash: How we use it

We started by testing it and see what types of logs we can manage with it. The initial setup was processing logs related to **dns, login, postresql, soap, epp, and system**.

We started using it in **production** and feeding logs from various sources.

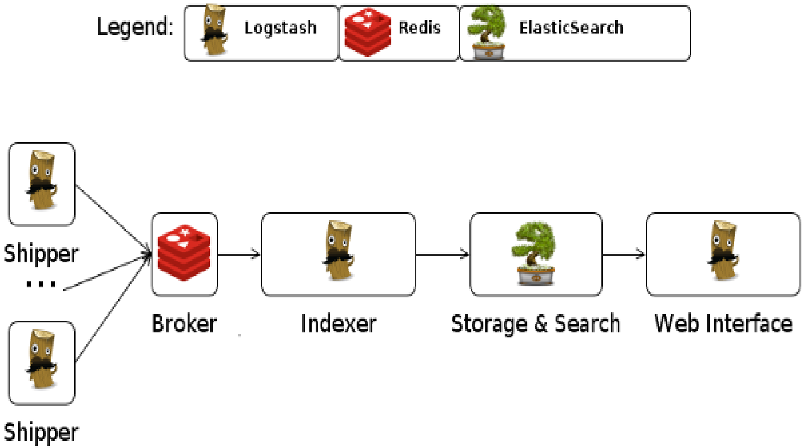
Logstash: Why we use it

We use logstash because I **to use its power in managing logs** and generate events based on specific triggers.

We cover states with **icinga and adding logstash** would cover the whole monitoring spectrum.

We like the fact that you can filter anything fast and simple with basically no programming skills.

Logstash: Overview of a centralized setup



Logstash: Hardware requirements

In order find out how much hardware you need for your logstash setup, first think **how much data you process daily** and if you need to search into it.

Another factor is the **complexity of your configuration**, specially filters.

In other words, there is no magic path/specific numbers (or at least I haven't found them).

How we use it with Elasticsearch

We have a standard elasticsearch cluster made out of 3 servers, storing aprox. 20GB per month.

We don't keep data more than 30 days.

We tend to use the latest stable for LS, ES and Kibana.

This can lead to problems from time to time because of rapid/major changes in these systems...

Logstash: How we use it with Kibana

We use it mainly for searching into logs

The graphing part is just for a quick overview/trend and not for storage.

I have some plans to integrate it more closely with icinga.

Logstash: Our deployment

Our current setup for LS is the following

1 centralized syslog server.
2 LS shippers sending logs to a redis server that acts as a queue.
10 LS indexers for various types of logs. Depending on the complexity of the logs I dedicate a whole indexer/process to one log type.
1 redis server that acts a the queue. Multiple queues running, depending how many log types I have.
That's it.

Logstash: Integrating with big data systems

In our case integrating with big data systems is done via **KairosDB**. This acts as a middleware and allows us to use as storage system **Cassandra**.

Sending from logstash can be done using the **cassandra** output.

Logstash: Using Kairosdb

Time series database written on top of Cassandra.

Fast and flexible.

Can push data via telnet, rest, graphite and other methods using custom plugins.

Logstash: Using Cassandra

Store more logs,graphs and keep them forever

The plan is to have the graphs more flexible than RRD's and don't loose precession over time.

Live datapoints and nice graphing options is a plus.

Logstash: Why Cassandra

Fast, simple, stable.

cqlsh, clustering, nodetool.

Fast, simple, stable.

Logstash: Graphing with Grafana

We use it to graph data from kairosdb.

Support for kairosdb has been added via a plugin.

Fast, simple, stable.

Logstash: Conclusion

In our case logstash seems to be a **great candidate** for covering the log management spectrum.

Its **ease of usage**, great collection of plugins and easy configuration makes it a great tool for managing logs.

It is definitely **a tool worth exploring** for this kind of monitoring.

Questions

This would be a great time to ask questions. :)

The end

Thank you!