Logstash and ElasticSearch deployment scenario at GSI

Matteo Dessalvi

HEPiX Fall 2013

Outline

- Monitoring at GSI.
- Entering Logstash.
- Dealing with logs in the UNIX way.
- Logstash + Redis + ElasticSearch.
- Use case scenarios.
- Demo time.
- Pros and cons of this solution.
- Further improvements
- Future developments.
- References.

Monitoring at GSI

- Nagios/Icinga
- Netdisco
- Collectd
- SNMP/MRTG/Torrus
- Logs analysis

So far, the last one it is the weakest aspect of our monitoring tools. It requires:

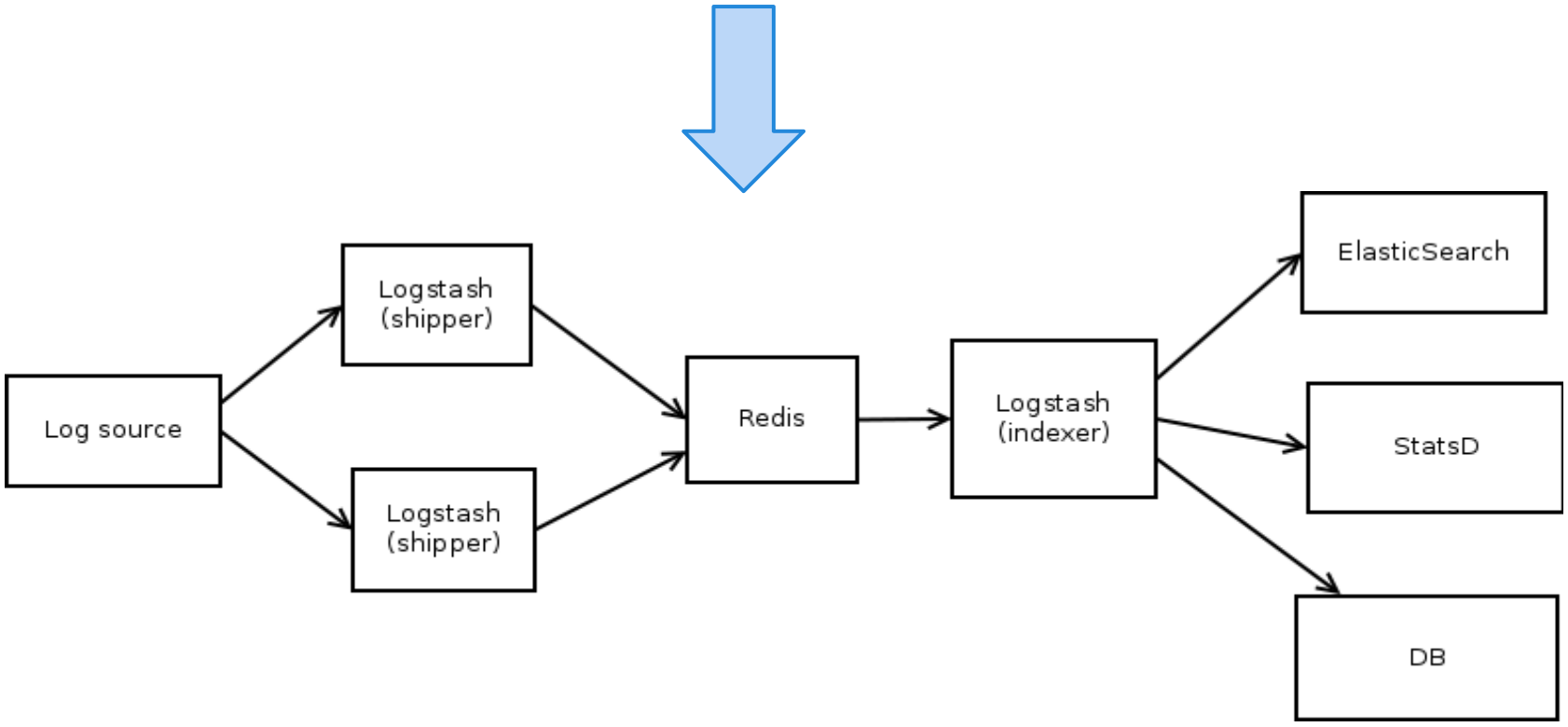
- *Custom shell/perl scripts*: lots of regex for different logs.
- *Summaries sent through email*: difficult to spot trends and look back at recurring events.
- *Search for specific events*: involve a lot of custom regex and time to search through gigabytes of text files.

Entering Logstash

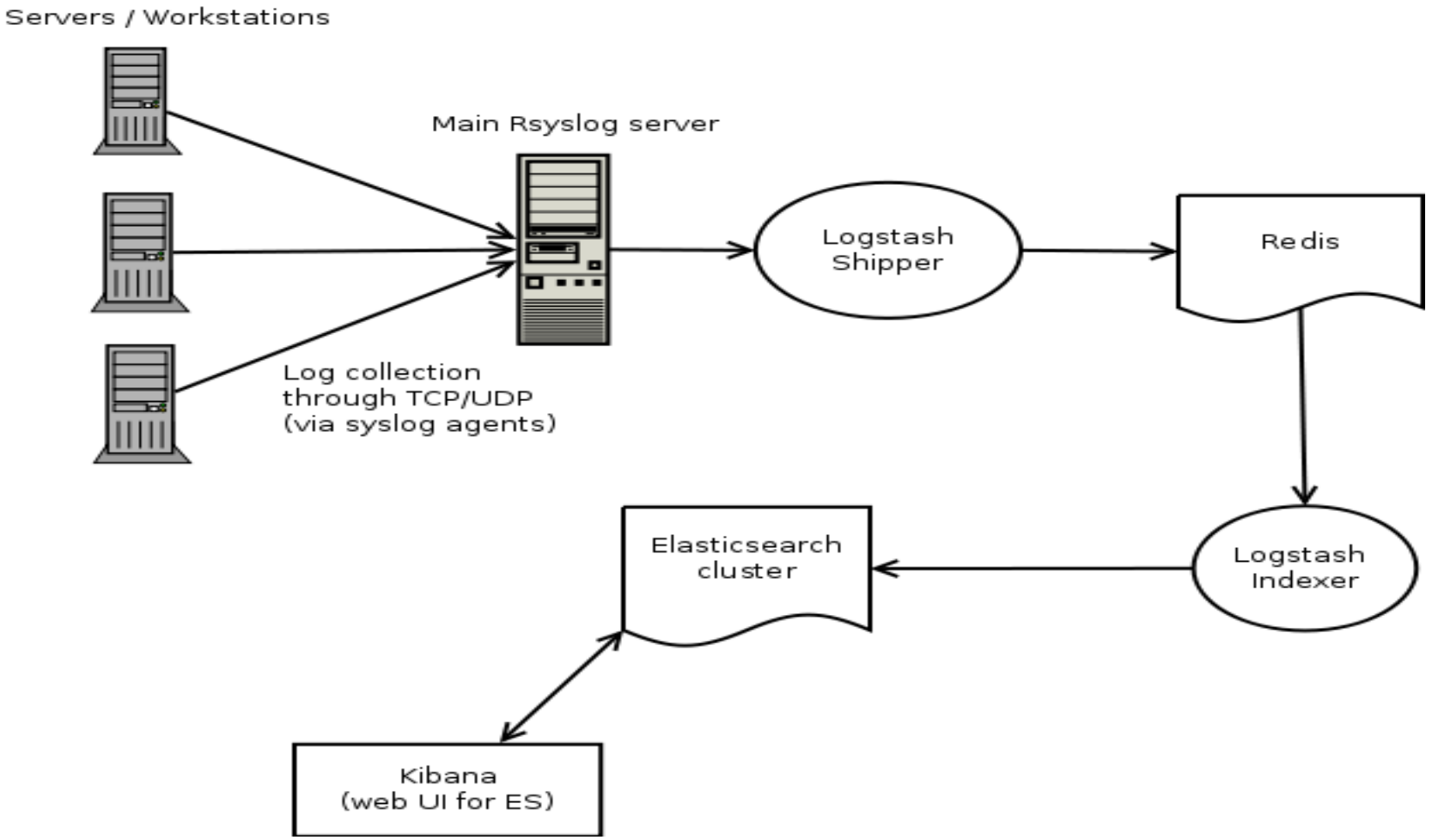
- *Logstash* is a tool for managing events and logs. You can use it to collect logs, parse them, and store them for later use.
- *Logstash* is written in **JRuby**, a Java implementation of Ruby, and it is available as a single JAR file.
- *Logstash* has input/filter/output plugins: "**Inputs** generate events, **filters** modify them and **outputs** ship them elsewhere".
- *Logstash* would not modify the usual way sysadmins look at log files. You can also think of it as "**UNIX pipes on steroids**".

Think about UNIX pipes

```
$ log_producer | grep … | sed … | awk … |
tee output | sort | uniq -c |
```



Logstash + Redis + ElasticSearch



Use cases

- Access logs from the SSH servers.
- Apache access logs.
- GridEngine accounting.
- Network devices logs (Cisco FWSM).
- Postfix MTAs logs.
- Email reports from Cfengine 2 (not completed yet).

Demo

It’s demo time!

Logstash for ZFS logs

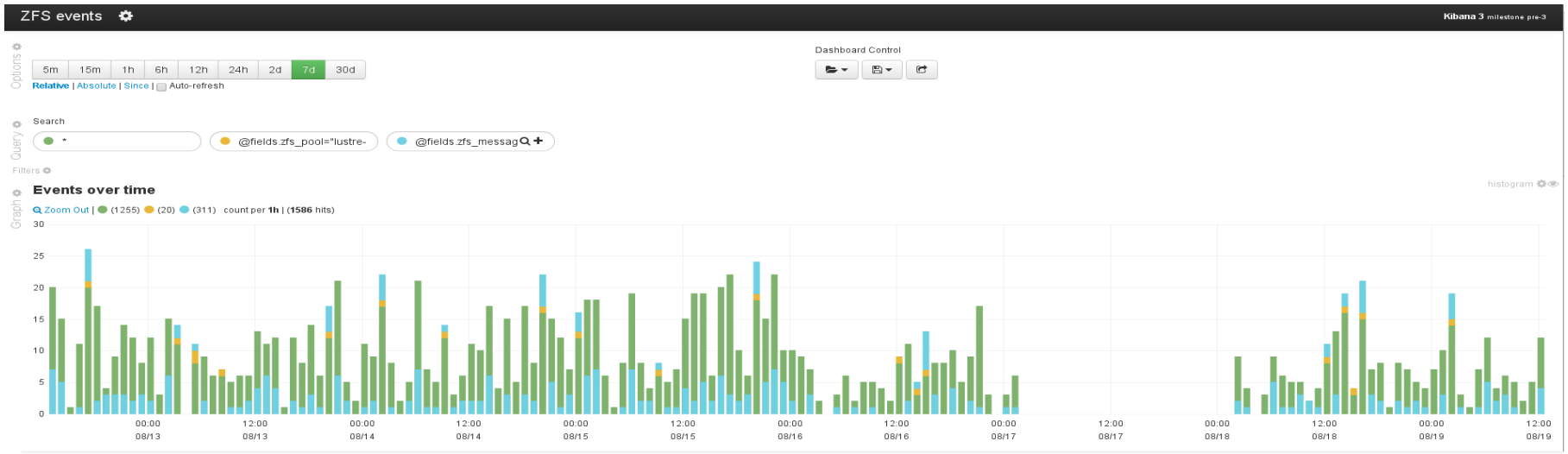
"*A first rudimentary ZFS event daemon*".

zeventd (autoreplace functionality):
https://github.com/zfsonlinux/zfs/issues/1763

NOTE: so far this daemon is only a proof of concept.
More info on: https://github.com/stibor/zfs

Currently GSI HPC is working on MIT Kerberos support for Lustre 2.x series.

ZFS event logs



Aug 19 10:41:28.397825305 +0000	lxlustr-zfs-616	zeventd	ereport.fs.zfs.scrub.start	lustre-zfs-pool-04
Aug 19 10:41:25.219684716 +0000	lxlustr-zfs-450	zeventd	resource.fs.zfs.statechange	lustre-zfs-pool-39
Aug 19 10:37:56.481870556 +0000	lxlustr-zfs-305	zeventd	ereport.fs.zfs.config.sync	lustre-zfs-pool-57
Aug 19 10:28:06.334772982 +0000	lxlustr-zfs-530	zeventd	ereport.fs.zfs.probe_failure	lustre-zfs-pool-21
Aug 19 10:21:19.081048904 +0000	lxlustr-zfs-162	zeventd	ereport.fs.zfs.scrub.start	lustre-zfs-pool-42
Aug 19 10:16:13.946141871 +0000	lxlustr-zfs-199	zeventd	resource.fs.zfs.statechange	lustre-zfs-pool-12
Aug 19 10:11:49.518705210 +0000	lxlustr-zfs-737	zeventd	ereport.fs.zfs.config.sync	lustre-zfs-pool-94
Aug 19 10:08:10.934396152 +0000	lxlustr-zfs-321	zeventd	ereport.fs.zfs.checksum	lustre-zfs-pool-17
Aug 19 10:02:16.362511807 +0000	lxlustr-zfs-480	zeventd	ereport.fs.zfs.scrub.start	lustre-zfs-pool-94
Aug 19 10:01:20.456716239 +0000	lxlustr-zfs-774	zeventd	ereport.fs.zfs.io	lustre-zfs-pool-60
Aug 19 09:41:46.536885133 +0000	lxlustr-zfs-851	zeventd	ereport.fs.zfs.probe_failure	lustre-zfs-pool-91
Aug 19 09:41:31.191753667 +0000	lxlustr-zfs-207	zeventd	ereport.fs.zfs.probe_failure	lustre-zfs-pool-47
Aug 19 09:17:54.472666110 +0000	lxlustr-zfs-886	zeventd	ereport.fs.zfs.config.sync	lustre-zfs-pool-55
Aug 19 09:05:34.077398014 +0000	lxlustr-zfs-399	zeventd	ereport.fs.zfs.probe_failure	lustre-zfs-pool-61
Aug 19 09:00:10.401885965 +0000	lxlustr-zfs-959	zeventd	ereport.fs.zfs.scrub.finish	lustre-zfs-pool-42
Aug 19 08:19:20.504048193 +0000	lxlustr-zfs-076	zeventd	ereport.fs.zfs.io	lustre-zfs-pool-96
Aug 19 08:09:13.521702289 +0000	lxlustr-zfs-179	zeventd	ereport.fs.zfs.io	lustre-zfs-pool-81
Aug 19 07:57:50.644058192 +0000	lxlustr-zfs-627	zeventd	ereport.fs.zfs.checksum	lustre-zfs-pool-19
Aug 19 07:49:41.748322168 +0000	lxlustr-zfs-013	zeventd	ereport.fs.zfs.scrub.start	lustre-zfs-pool-20
Aug 19 07:49:32.516666200 +0000	lxlustr-zfs-980	zeventd	ereport.fs.zfs.checksum	lustre-zfs-pool-48
Aug 19 07:40:11.294787784 +0000	lxlustr-zfs-132	zeventd	ereport.fs.zfs.io	lustre-zfs-pool-52
Aug 19 07:02:43.302302995 +0000	lxlustr-zfs-100	zeventd	ereport.fs.zfs.probe_failure	lustre-zfs-pool-14
Aug 19 06:51:28.272205420 +0000	lxlustr-zfs-990	zeventd	ereport.fs.zfs.scrub.finish	lustre-zfs-pool-04
Aug 19 06:47:55.424442965 +0000	lxlustr-zfs-702	zeventd	ereport.fs.zfs.scrub.start	lustre-zfs-pool-65
Aug 19 06:39:11.600390373 +0000	lxlustr-zfs-188	zeventd	ereport.fs.zfs.scrub.start	lustre-zfs-pool-18
Aug 19 06:25:52.475372203 +0000	lxlustr-zfs-373	zeventd	ereport.fs.zfs.probe_failure	lustre-zfs-pool-95
Aug 19 06:20:35.103083601 +0000	lxlustr-zfs-850	zeventd	ereport.fs.zfs.checksum	lustre-zfs-pool-14
Aug 19 06:09:46.298369455 +0000	lxlustr-zfs-949	zeventd	ereport.fs.zfs.scrub.start	lustre-zfs-pool-96
Aug 19 05:41:32.352987485 +0000	lxlustr-zfs-096	zeventd	ereport.fs.zfs.config.sync	lustre-zfs-pool-93
Aug 19 05:41:05.182222571 +0000	lxlustr-zfs-341	zeventd	ereport.fs.zfs.scrub.start	lustre-zfs-pool-20
Aug 19 05:04:36.889262313 +0000	lxlustr-zfs-633	zeventd	ereport.fs.zfs.scrub.start	lustre-zfs-pool-53
Aug 19 05:03:34.997076937 +0000	lxlustr-zfs-604	zeventd	ereport.fs.zfs.probe_failure	lustre-zfs-pool-45

Kibana 3 vs Kibana 2

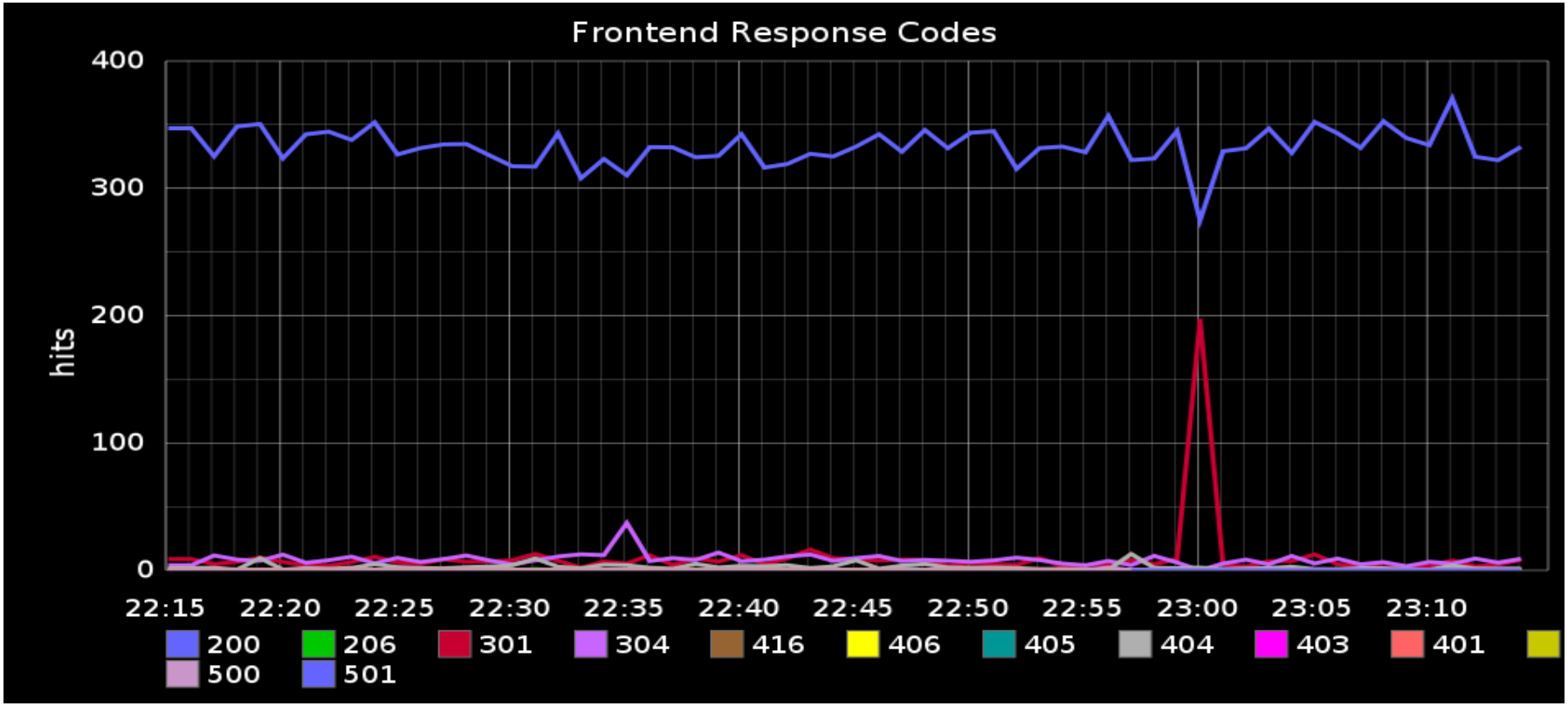
As you have seen on the demo there are some differences between the ‘old’ version of Kibana and the new one.

The most relevant changes are:

- Completely rewritten: from *Ruby* to *HTML* + *Javascript (AngularJS)*.
- A bare bones web server can serve it **but** you need to carefully proxy the access to ElasticSearch.
- The load burden will be on the client side and no more on the server side.

Further improvements

- *Graphite and Statsd*: this solution will add an additional layer to the information gathered from the logs.



Future developments

Migration from Logstash 1.1.x to 1.2.x.

- Increased performances, especially if you have to deal with a lot of filters.
- Conditionals support in the filter section.
- Support for the Bulk API of ElasticSearch.

Mandatory requirement: rethink the ElasticSearch templates applied to your index.

Pros and cons

- Filtering capabilities will let you be able to deal with different kind of log files.
- Logs and events can be collected from nearly any kind of input sources: TCP/UDP, Syslog, scripts or commands output, DBMS, etc.

- ElasticSearch is not meant to be a long term storage tool.
- Security is still a relevant problem: so far it is not possible to define access roles (possible solution: https://github.com/sonian/elasticsearch-jetty)
- JVM options need to be tuned carefully.

References

Logstash, ElasticSearch and Kibana:

http://logstash.net/

http://www.elasticsearch.org/overview/

http://www.elasticsearch.org/overview/kibana/

Grok patterns debug:

http://grokdebug.herokuapp.com/

Logstash 1.2.2 released:

https://github.com/logstash/logstash/blob/master/CHANGELOG#L1-L121

Highlighting differences between LS 1.1.x and 1.2.x:

http://tobrunet.ch/2013/09/logstash-1-2-0-upgrade-notes-included/

Thank you!

Questions?