Centralised Logging with Logstash and Kibana

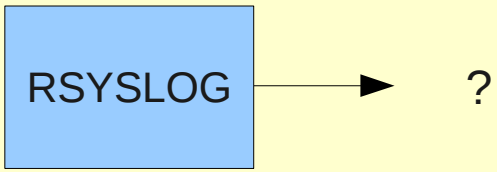
(and rsyslog,
and elasticsearch,
and ...)

Matthew Richardson
(Engineering)
18th January 2013

Centralised Logging?

Rsyslog

- Default syslogd in LCFG
- Easy to log centrally
- Disk Buffering for safety



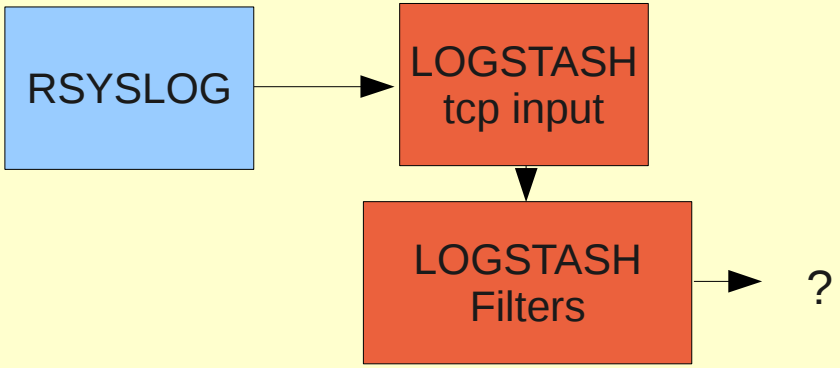

```
$ActionQueueType LinkedList

$ActionQueueFileName
LogstashBuffer

$ActionResumeRetryCount -1

$ActionQueueSaveOnShutdown
on

$ActionQueueMaxDiskSpace 1G

*.* @@log.see.ed.ac.uk:5544
```

Logstash

Inputs (25+)	Filters (20+)	Outputs (35+)
file	grep	file
tcp/udp	grok	tcp/udp
XMPP (Jabber)	mutate	elasticsearch
log4j	anonymize	XMPP (Jabber)
stdin	dns	email
Windows eventlog	XML	ganglia
	metrics	graphite
...	...	...

Logstash Configuration

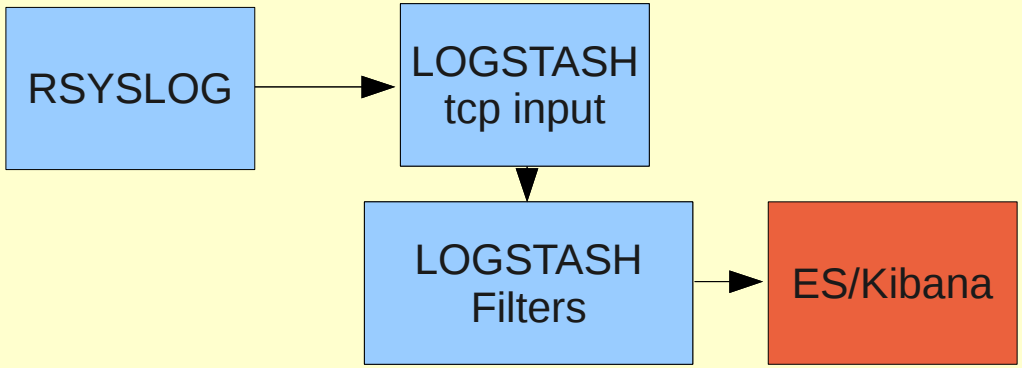
- TCP/UDP input
- Types and Tagging
- Powderful Filtering
- Structured Output

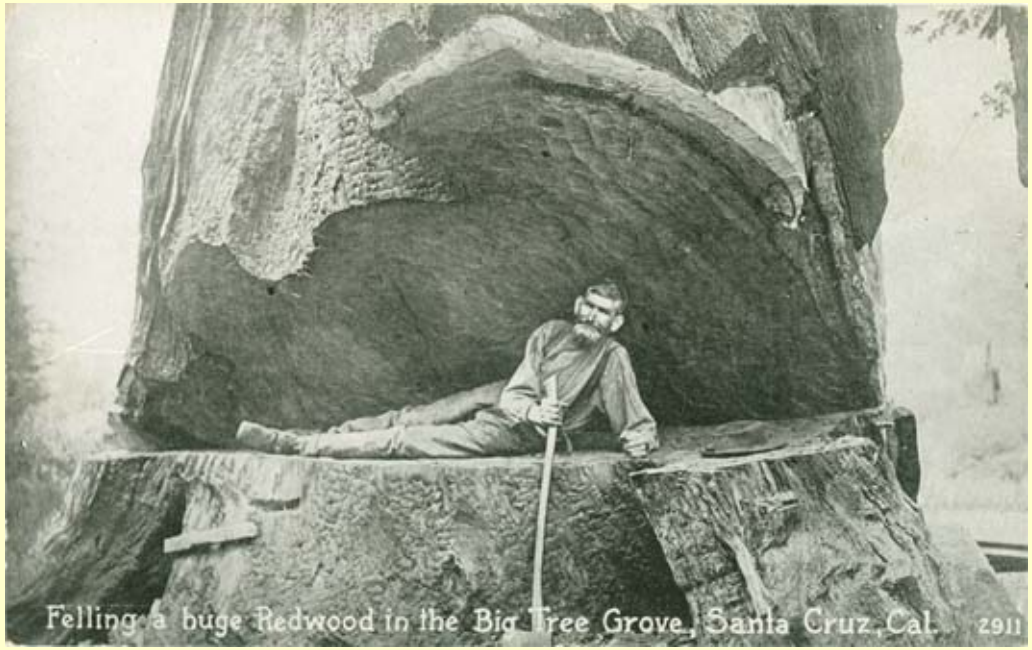



```
input {
  tcp {
    port => 5544
    type => syslog
  }
}
filter {
  grok {
    pattern => [ "<%{POSINT:syslog_pri}>
%{SYSLOGTIMESTAMP:syslog_timestamp}
%{SYSLOGHOST:syslog_hostname}
%{PROG:syslog_program}(?:\[
%{POSINT:syslog_pid}\])?:
%{GREEDYDATA:syslog_message}" ]
  }
  syslog_pri { type => "syslog" }
  date { match => [ “syslog_timestamp”,
                    "MMM  d HH:mm:ss",
                    "MMM dd HH:mm:ss" ]
       }
  mutate {
      exclude_tags => "_grokparsefailure"
      replace => [ "@message", "%{syslog_message}" ]
  }
  mutate {
      remove => [ "syslog_message",
"syslog_timestamp" ]
  }
}
output { elasticsearch { embedded => false }}
```

Log Analysis

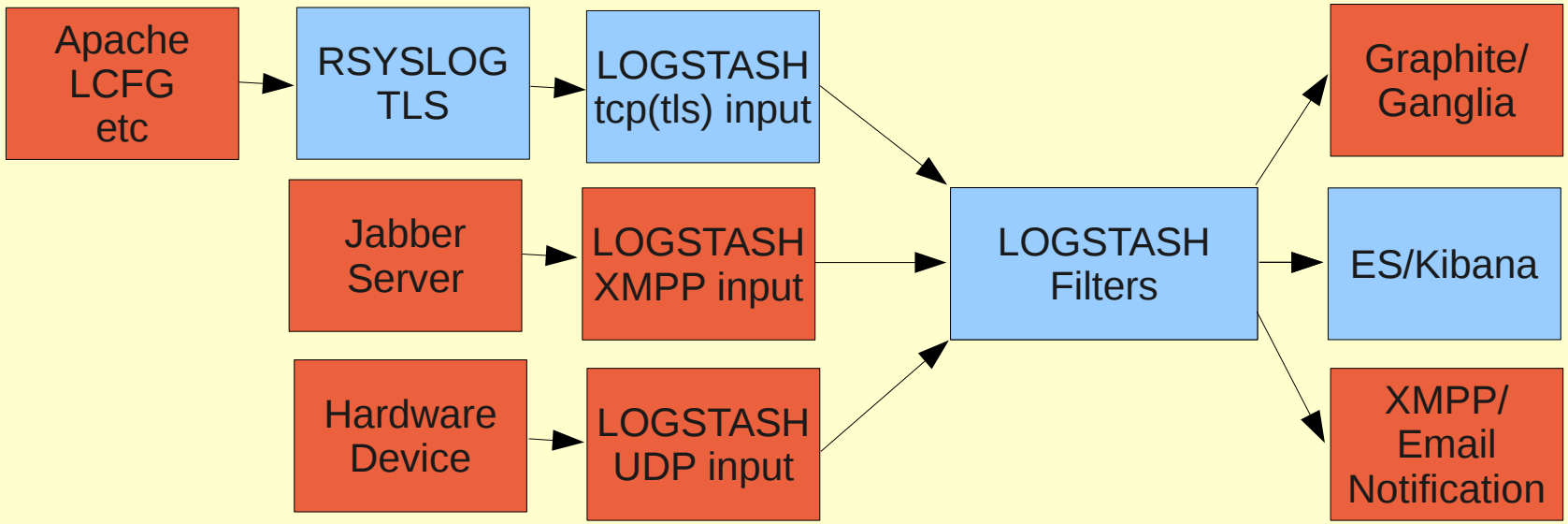
- Elasticsearch backend
- Kibana web frontend





Future Plans

- Encryption/Authentication for logging
- Handling other (non-syslog) logs
- Other Inputs
- Metrics



Questions?

m.richardson@ed.ac.uk
Jabber(dst): mrichar1