

Centralised logfile analysis using Elastic, Logstash and Kibana (ELK)

Tomas Krajca

<tomas@repositpower.com>

PUG Meetup
Canberra

June 2, 2016

Today

- 1 Reporting state of (python) programs.
- 2 Storing these reports.
- 3 Processing/indexing these reports.
- 4 Searching/analyzing these reports.
- 5 ELK pipelines.
- 6 Alerting from these reports.
- 7 Lessons learned.
- 8 QA.

Reporting state of (python) programs

Why and how?

- 1 example1.py
- 2 example2.py
- 3 example3.py
- 4 example4.py
- 5 logging.ini
- 6 <https://docs.python.org/2/library/logging.html>
- 7 <https://docs.python.org/2/library/logging.config.html>

Storing these reports

Log storage

- 1 File
- 2 AWS S3
- 3 Console
- 4 Rsyslog
- 5 Heroku
- 6 ...

Processing/indexing these reports.

Logs

```

2016-06-02 00:21:06,705 DEBUG connectionpool requests.packages
2016-06-02 00:21:06,706 DEBUG worker worker0.55 PID: 1360 Res
2016-06-02 00:21:06,707 INFO master stsd_db_proxy.master PID: 12
2016-06-02 00:21:07,910 DEBUG master stsd_db_proxy.master PID: 1
2016-06-02 00:21:07,910 INFO master stsd_db_proxy.master PID: 12
2016-06-02 00:21:07,962 DEBUG worker worker0.55 PID: 1360 POS
2016-06-02 00:21:07,972 DEBUG connectionpool requests.packages
2016-06-02 00:21:07,974 INFO master stsd_db_proxy.master PID: 12
2016-06-02 00:21:07,998 DEBUG base zmq.auth PID: 1230 version
2016-06-02 00:21:07,998 DEBUG base zmq.auth PID: 1230 ALLOWED
2016-06-02 00:21:07,998 DEBUG zmq_ zmq.auth PID: 1230 Authent
2016-06-02 00:21:07,998 DEBUG base zmq.auth PID: 1230 ZAP rep
...

```


Logstash

- 1 Tool that crunches logs/reports.
- 2 Configurable to process any sorts of logs.
- 3 Ruby/java.
- 4 Collect - Process - Forward events/messages.
- 5 Input(s) - Filter(s) - Output(s).
- 6 <https://www.elastic.co/products/logstash>

Logstash inputs

```
input {  
  file {  
    type => "python"  
    path =>  
      ["/var/log/apps/app.stderr.log"]  
    add_field => [ "program", "app" ]  
    sincedb_path =>  
      "/var/lib/logstash/sincedb"  
  }  
}
```

Logstash filters

- 1 logstash.conf
- 2 <https://github.com/elastic/logstash/blob/v1.4.2/patterns/grok-patterns>
- 3 \$ logstash -v -f logstash.conf

Logstash filters

```
{
  "message" => "2016-06-01 12:52:41,129 DEBUG base zmq.auth
                PID: 1230 ZAP reply code=200 text=OK",
  "@version" => "1",
  "@timestamp" => "2016-06-01T12:52:41.129Z",
  "host" => "dauvmfes001.repositpower.net",
  "path" => "/var/log/apps/SecureTSDBProxy.stderr.log",
  "type" => "python",
  "program" => "SecureTSDBProxy",
  "loglevel" => "DEBUG",
  "module" => "base",
  "logger" => "zmq.auth",
  "received_at" => "2016-06-01T12:52:41.182Z",
  "received_from" => "dauvmfes001",
  "@source_host" => "dauvmfes001",
  "@message" => "ZAP reply code=200 text=OK",
  "tags" => [
    [0] "_parsed"
  ]
}
```

Searching/analyzing these reports

Logstash outputs

- ① Zabbix
- ② Redis
- ③ Elasticsearch
- ④ ...

```
output {
  if "zabbix" in [tags] {
    zabbix {
      host => "zabbix.repositpower.net"
      port => "10051"
      zabbix_sender =>
        "/usr/bin/zabbix_sender_retry"
    }
  }
  elasticsearch {
    host => "search.repositpower.net"
    protocol => "http"
    flush_size => 1000
    max_retries => 7
  }
}
```

Elasticsearch

- 1 NoSQL database.
- 2 Scalable, robust, powerful.
- 3 Full-text search server built on Lucene indexes.
- 4 (JSON) Schema free.
- 5 HTTP/JSON or TCP API/interface
- 6 Java
- 7 Indexes - Databases
- 8 <https://www.elastic.co/>

Elasticsearch demo

- 1 Python client libraries.
- 2 `https://pypi.python.org/pypi/elasticsearch/2.3.0`
- 3 `http://search.repositpower.net:9200/`
- 4 `http://search.repositpower.net:9200/_cat/indices?v`
- 5 `curl -d "'loglevel': 'INFO', '@message': 'Tomas Krajca2'"`
`http://search.repositpower.net:9200/test/2/_update`
- 6 `http://search.repositpower.net:9200/logstash-2016.06.02/_search?q=@message:"timedout"&pretty=true`
- 7 `$ curator show indices --all-indices`

Kibana

- ① Data visualization and search for elasticsearch.
- ② <https://www.elastic.co/products/kibana>

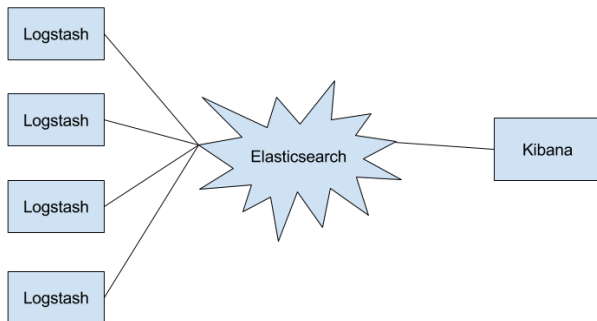
Kibana demo

- ① `time_duration:[3000TO*]ANDhttp_request:*`
- ② `bytes_read:[3000TO*]ANDhttp_request:*`
- ③ `loglevel:ERRORORloglevel:WARNINGAND@message:timeoutANDNOT@message:Unable`

ELK pipelines

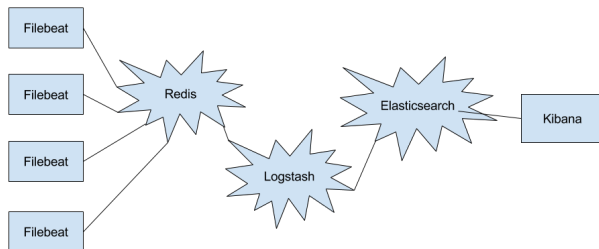
Distributed

① Logstash + Elasticsearch + Kibana



Centralized

- 1 Logstash + Redis + Logstash + Elasticsearch + Kibana
- 2 Logstash forwarder/Filebeat + Redis + Logstash + Elasticsearch + Kibana
- 3 Rsyslog forwarder + Rsyslog master + Logstash + Elasticsearch + Kibana



Considerations

- 1 Complexity.
- 2 How much logging on each node.
- 3 How strict SLAs for each service.
- 4 How reliable should it be.
- 5 How fast/real-time should it be.
- 6 Who does the hard work.

Alerting from these reports

Bosun

- ① Monitoring and alerting system by Stack Exchange.
- ② Flexible language to define alerts and notifications.
- ③ Supports OpenTSDB, Graphite, Elasticsearch, Influx, ... as backend dbs.
- ④ <https://bosun.org>

Bosun demo

- 1 bosun.conf
- 2 `max(lsstat("logstash", "backend_name, server_name",
"program:haproxy, http_request:
.*", "time_duration", "max", "10s", "5m", ""))>5000`
- 3 `lscount("logstash", "@source_host, program", "loglevel:
error", "60s", "3h", "")`
- 4 `lscount("logstash", "server_name, backend_name", "program:
haproxy, http_request:.*", "1h", "3d", "")`
- 5 Response time alert in rule editor
- 6 Anomaly-based alerting, alert aggregation.

Lessons learned

Tips & tricks

- 1 Python logging with multiprocessing - avoid files.
- 2 Advantages of non-blocking/async logging.
- 3 Don't use multiline logstash input codec.
- 4 HW resources for logstash and elasticsearch.
- 5 Event queuing in logstash when elasticsearch is down.
- 6 Consensus on meaning of log levels for alerting.
- 7 Educate staff on using kibana/tools.
- 8 Never silence python stacktraces.
- 9 SIGUSR1/SIGUSR2/setupLogging() hooks for dropping logging level, dumping threads and more :).
- 10 Correlation IDs.

Questions/discussion

Thank you

tomas@repositpower.com

<https://github.com/tkrajca/PUG-ELK>