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 setups.
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 stsdb_proxy.master PID: 12
2016-06-02 00:21:07,910 DEBUG master stsdb_proxy.master PID: 1
2016-06-02 00:21:07,910 INFO master stsdb_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 stsdb_proxy.master PID: 12
2016-06-02 00:21:07,998 DEBUG base zmq.auth PID: 1230  versio
2016-06-02 00:21:07,998 DEBUG base zmq.auth PID: 1230  ALLOWE
2016-06-02 00:21:07,998 DEBUG zmq_ zmq.auth PID: 1230  Authen
2016-06-02 00:21:07,998 DEBUG base zmq.auth PID: 1230  ZAP re
...
```

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

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:"timed out"&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:[3000 TO *] AND http_request:*
2. bytes_read:[3000 TO *] AND http_request:*
3. loglevel:ERROR OR loglevel:WARNING AND @message:timeout AND NOT @message:Unable

ELK setups

Distributed

Centralized

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

Alerting from these reports

Bosun

1. response time alert
2. too many warnings/errors alert
3. anomaly alert (number of errors)

Lessons learned

Tips & tricks

1. python logging with multiprocessing
2. advantages of async logging
3. multiline logstash codec being stuck/old alerts
4. HW resources for logstash and elasticsearch
5. queuing in logstash when elaslticsearch is down
6. consensus on meaning of log levels for alerting
7. educate on how to use kibana
8. bosun very young/rough
9. python stacktraces in the logs
10. SIGUSR1/SIGUSR2
11. logstash testing/debugging (rubydebug)

Questions/discussion

Thank you

tomas@repositpower.com
https://github.com/tkrajca/PUG-ELK