HANDS-ON ELK WORKSHOP

AGENDA FOR TODAY

17:30-18:00	Intro
18:00-18:30	Configuration (Install ELK + repo)
18:30-19:10	1st pipeline logstash + Elasticsearch
19:10-19:20	pause
19:20-19:50	1st pipeline visualization in Kibana
19:50-20:00	Pause
20:00-20:50	2nd pipeline logstash + Elasticsearch
20:50-21:00	Pause
21:00-21:40	2nd pipeline visualization in Kibana

WHAT IS THE ELK PLATFORM?

ELK consist of three open source projects — Elasticsearch, Logstash, and Kibana — designed to take data from any source and search, analyze, and visualize it in real time. The philosophy behind these tools is that getting immediate, actionable insight from data matters.

- **Elasticsearch** for deep search and data analytics.
- **Logstash** for centralized logging management: shipping and forwarding logs, log enrichment, and parsing.
- **Kibana** for powerful and beautiful data visualizations.

WHAT CAN WE USE ELK FOR?

- Issue debugging
- Performance analysis
- Security analysis
- Predictive analysis
- Internet of things (IoT) and logging

TYPICAL PROBLEMS WITH YOUR LOGS

- Non-consistent log format
- Decentralized logs
- Expert knowledge requirement

SETUP

- Clone https://github.com/omrisiri/devops/
- run setup.sh to download and extract in the elk-workshop directory in the repository:
  - Elasticsearch
  - Logstash
  - Kibana
- Edit elasticsearch-2.1.1/config/elasticsearch.yml
  - cluster.name: ${HOSTNAME}

SHELL

- Start a shell in <elk-workshop>/elasticsearch-2.1.1/bin
  - Run elasticsearch
  - Open your browser at http://127.0.0.1:9200/
- Start a shell in <elk-workshop>/kibana-4.3.1-xxx/bin
  - Run kibana
  - Open your browser at http://127.0.0.1:5601/
- Start a shell in <elk-workshop>/logstash/pipelines/setup
  - This will be used for logstash

VERIFY LOGSTASH

- Follow instructions at setup.txt

Linux:

```
../../../logstash-2.1.1/bin/logstash agent -f verify.conf --configtest
```

Windows:

```
..\..\..\logstash-2.1.1\bin\logstash agent -f verify.conf --configtest
```

PIPELINES

- LAPD Crime Reports
- HTTP Access Logs

LAPD CRIME REPORTS

- Navigate to **./logstash/pipelines/lapd**
- Familiarize yourself with the data **./logstash/pipelines/lapd/data/lapd_small.csv**
- We will focus on the following headers:

DATE OCC	Date of occurrence
TIME OCC	Time of occurrence
Crm Cd	Crime Code
Crm Cd Desc	Crime Code Description
Status	
Statue Desc	
LOCATION	Street address
location 1	GPS coordinates

1ST STEP: READ THE DATA

What:	Learn how to use the file input plugin
How:	Open 1.txt and roll up your sleeves
When:	Now. You have 3 minutes!

Ærg help! https://www.elastic.co/guide/en/logstash/current/plugins-inputs-file.html

2ND STEP: GIVE STRUCTURE TO THE DATA

What:	Familiarize yourself with the csv filter plugin
How:	Open 2.txt and read.
When:	Now. You have 5 minutes!

Ærg help!

https://www.elastic.co/guide/en/logstash/current/plugins-filters-csv.html

3RD STEP: CLEAN AND FORMAT THE DATA

What:	Familiarize yourself with mutate and date filter plugins
How:	Open 3.txt
When:	Now. You have 5 minutes!

Ærg help!

https://www.elastic.co/guide/en/logstash/current/plugins-filters-mutate.html

https://www.elastic.co/guide/en/logstash/current/plugins-filters-date.html

4TH STEP: EXPORT DATA TO ELASTICSEARCH

What:	Familiarize yourself with elasticsearch output plugin
How:	Open 4.txt
When:	Now. You have 5 minutes!

Ærg help!

https://www.elastic.co/guide/en/logstash/current/plugins-outputs-elasticsearch.html

KIBANA VISUALIZATION

- Settings tab
  - Get lapd index
- Discover tab
  - Play with the time filter
  - See the structure of the data
- Visualize tab
  - Generate Pie charts
  - Histogram bars
  - Line charts for trends
  - Metrics
  - Filter aggregations
- Dashboard tab
  - Construct a dashboard
  - How to import / export the dashboard

HTTP ACCESS LOGS

Access logs generated by a script based on:
https://gist.github.com/fetep/2037301

Logs, exercises and configuration files can be found in
logstash/pipelines/httpd

GROK

- Regular expression text parser
- Pre-defined patterns
  - See: https://github.com/logstash-plugins/logstash-patterns-core/
- Named matches become fields

GETTING STARTED

- Have a look at **data/access.mini.log**
- Adapt the paths in **1.conf**
- Run logstash and take note of the **test** field:

Windows:

```
..\..\..\logstash-2.1.1\bin\logstash agent -f 1.conf
```

Linux:

```
../../../logstash-2.1.1/bin/logstash agent -f 1.conf
```

MATCH OPTION

- Take note of the pattern used: "%{DATA:test} "
- **DATA** is a pre-defined pattern equivalent to ".*?"
- **:test** tells grok to bind the match to the field **test**
- "%{DATA:test} " is equivalent to "(?<test>.*?) "

GROK CONSTRUCTOR

- Regular expressions can be a hassle
- Lots of pre-defined patterns (around 120): https://github.com/logstash-plugins/logstash-patterns-core/
- http://grokconstructor.appspot.com/ to the rescue

INCREMENTAL CONSTRUCTION

- Select incremental construction
- Copy a few lines from access.mini.log into the text area and press Go
- Notice that the first pattern in the list matches everything: **COMBINEDAPACHELOG**
  - In the final results, we will use this pattern.
    For now, spend a few minutes getting familiar with the constructor.

INCREMENTAL CONSTRUCTION CONT.

- The Apache log format documentation: https://httpd.apache.org/docs/1.3/logs.html#common
- Try to build a pattern that will capture the following fields:
  - Client IP/host name
  - Date and time
  - HTTP method
  - Path part of requested URL
  - HTTP status code
- Feel free to handle more parts
- Remember to add field names to the pattern
- Test your patterns

GEO IP

- Adds GPS coordinates based on IP addresses.
- A database mapping IP addresses to cities is included in logstash.
- Updated databases can be downloaded from http://dev.maxmind.com/geoip/legacy/geolite/

BASIC GEO IP CONFIGURATION

- Use **2.conf**, or add a geoip filter after your grok filter
- First set the source field to the client IP/host name field
- You can find the field by examining the COMMONAPACHELOG pattern
  or by running the configuration before adding the geoip filter
- Try running logstash with the configuration

FIELDS

- The geoip has added a lot of fields
- The most important one is [**geoip**][**location**] (coordinates)
- All these fields take up additional storage space
- Add a **fields** option to the geoip filter and specify a string array of fields you want to keep
- Re-run logstash with the updated configuration

TIMESTAMP

- Use **3.conf**, for this and the next exercise
- Format specification can be found at: http://joda-time.sourceforge.net/apidocs/org/joda/time/format/DateTimeFormat.html
- Add a date filter similar to the one used in the LAPD exercise
- You don't need to specify the time zone, because the Apache date format contains it

CHECKSUM

- Add a checksum with the checksum filter: https://www.elastic.co/guide/en/logstash/current/plugins-filters-checksum.html
- Set the algorithm to sha256 (default) or md5
- Set the keys to use the **message** field only
- You cannot specify the output field, so we move it with a mutate
  - Add a [**@metadata**][**computed_id**] field with the value of the **logstash_checksum** field
  - Remove the **logstash_checksum** field

OUTPUT TO ELASTICSEARCH

- Add output to Elasticsearch
- Set the name of the index

IMPORT FULL ACCESS LOG

- Unzip the **data/access.zip** archive
- Run logstash with the final configuration

WRAP-UP

USEFUL LINKS

- Follow the blog https://www.elastic.co/blog
- Some books
  - https://www.packtpub.com/big-data-and-business-intelligence/elasticsearch-cookbook
  - https://www.packtpub.com/big-data-and-business-intelligence/learning-elk-stack

UNIT/INTEGRATION TESTS

- Testing Logstash configurations can be difficult
- It is possible to write unit tests in Ruby:
- http://stackoverflow.com/questions/18823917/how-to-implement-the-unit-or-integration-tests-for-logstash-configuration

TIME-BASED INDICES

- You can add date fields to the index name
  - Slight increase in storage requirements
  - Allows deleting partial data, which saves storage
  - Increased performance?
- You may want indices to be:
  - Daily: "-%{+YYYY.MM.dd}"
  - Weekly "-%{+xxxx.ww}"
  - Monthly "-%{+YYYY.MM}"
- Defaults to daily: "logstash-%{+YYYY.MM.dd}"