How about no grep and zabbix?

or

ELK based metrics and alerts.

Vladimir Pavkin

QIWI Conf #3

Scala Developer

gettopical.com

Plan

- ELK stack components
- ELK setups
- Basic usage (grep)
- Advanced usage (alerts, metrics)

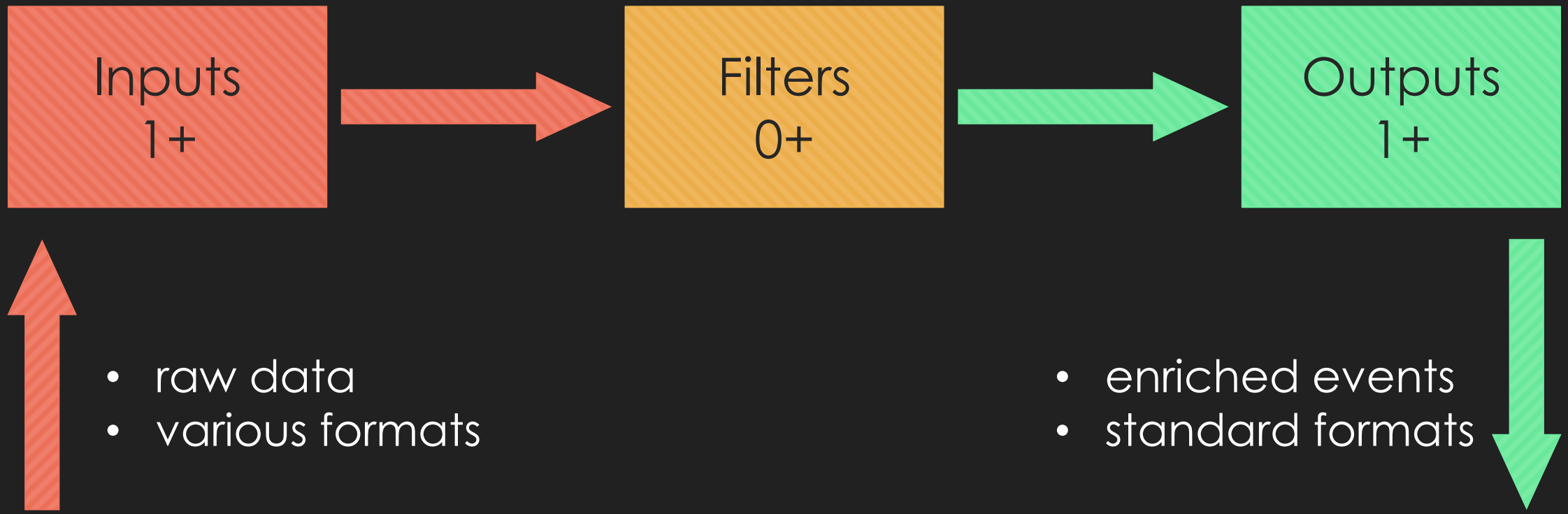
Da heck is ELK?

Elasticsearch

Logstash

Kibana

Logstash


Inputs
1+
Filters
0+
Outputs
1+
• raw data
• various formats
• enriched events
• standard formats

Logstash inputs

- file, syslog, log4j
- http, tcp, udp
- rabbitmq
- twitter, rss
- …

Logstash filters

- grok
- geoip
- drop, clone, mutate, split
- …

Logstash outputs

- stdout, file
- elasticsearch
- rabbitmq
- http, tcp, udp
- …

Logstash transformation example

```
127.0.0.1 - - [11/Dec/2013:00:01:45 -0800]
"GET /xampp/status.php HTTP/1.1" 200 3891
"http://cadenza/xampp/navi.php" "Mozilla/5.0 (Macintosh;
Intel Mac OS X 10.9; rv:25.0) Gecko/20100101 Firefox/25.0"
```

Logstash transformation example

```
{
  "message" : "127.0.0.1 - - [11/Dec/2013:00:01:45 -0800] \"GET /xampp/status.php HTTP/1.1\" 200 3891 \"http://cadenza/xampp/navi.php\" \"Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:25.0) Gecko/20100101 Firefox/25.0\"",

  "@timestamp" : "2013-12-11T08:01:45.000Z",
  "@version" : "1",
  "host" : "cadenza",
  "clientip" : "127.0.0.1",
  "path" : "/xampp/status.php",
  "response" : "200",

  // ...

}
```

Elasticsearch

Distributed data storage, optimized for text search.

Is managed/queried through HTTP REST API.

Based on open-source Apache Lucene project.

Elasticsearch vocab

- **Document** - data unit used for indexing and searching. Contains **fields**. (JSON)
- **Field** - a part of a **Document**, has a **name** and a **value**.
- **Term** - the unit of search. A word we look for in text

Elasticsearch inverted index

Documents:

1 => How about no grep

2 => How about no zabbix

Index:

```
how     => <1>, <2>
about   => <1>, <2>
zabbix  => <2>
...
```

Elasticsearch query DSL

- message:exception
- log_level:(ERROR OR INFO)
- stack_trace:”Out of memory”
- _exists_:body
- date:[now-1h TO now]
- age:(>=10 AND <20)

- wildcards (*, ?)
- regex search
- fuzzy search (Levenshtein)
- proximity search
- boosting weights

Elasticsearch query DSL

Even more power with JSON query DSL:

```
{
  "query_string" : {
    "fields" : [“message", “stack_trace"],
    "query" : “exception AND fatal"
  }
}
```

Elasticsearch clustering

Just easy as hell

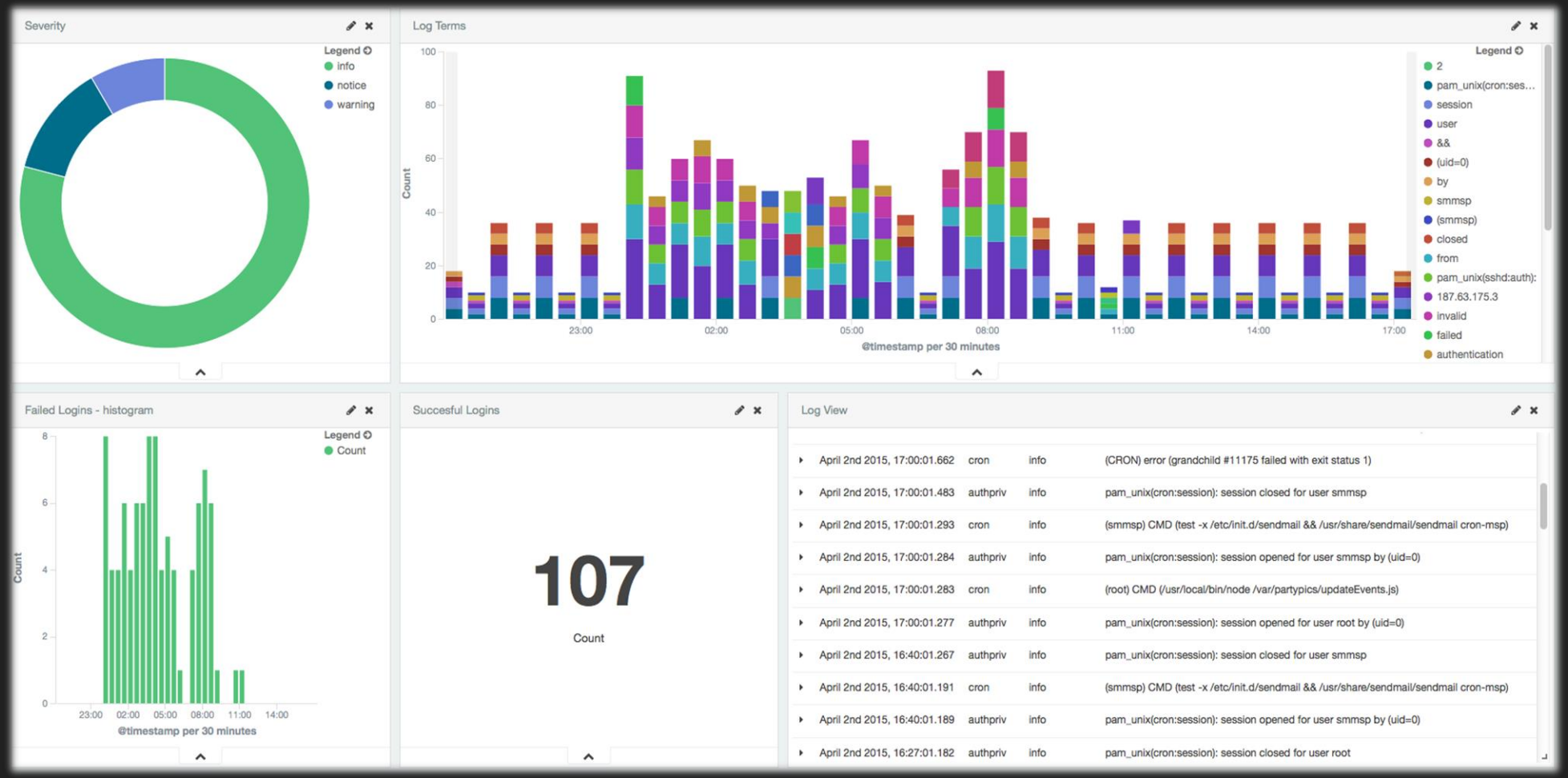
Kibana

Converts these:

```
{
  "message" : "127.0.0.1 - - [11/Dec/2013:00:01:45 -0800] \"GET /xampp/status.php
HTTP/1.1\" 200 3891 \"http://cadenza/xampp/navi.php\" \"Mozilla/5.0 (Macintosh; Intel
Mac OS X 10.9; rv:25.0) Gecko/20100101 Firefox/25.0\"",
  "@timestamp" : "2013-12-11T08:01:45.000Z",
  "@version" : "1",
  "host" : "cadenza",
  "clientip" : "127.0.0.1",
  "path" : "/xampp/status.php",
  "response" : "200",
  // ...
}
```

Kibana

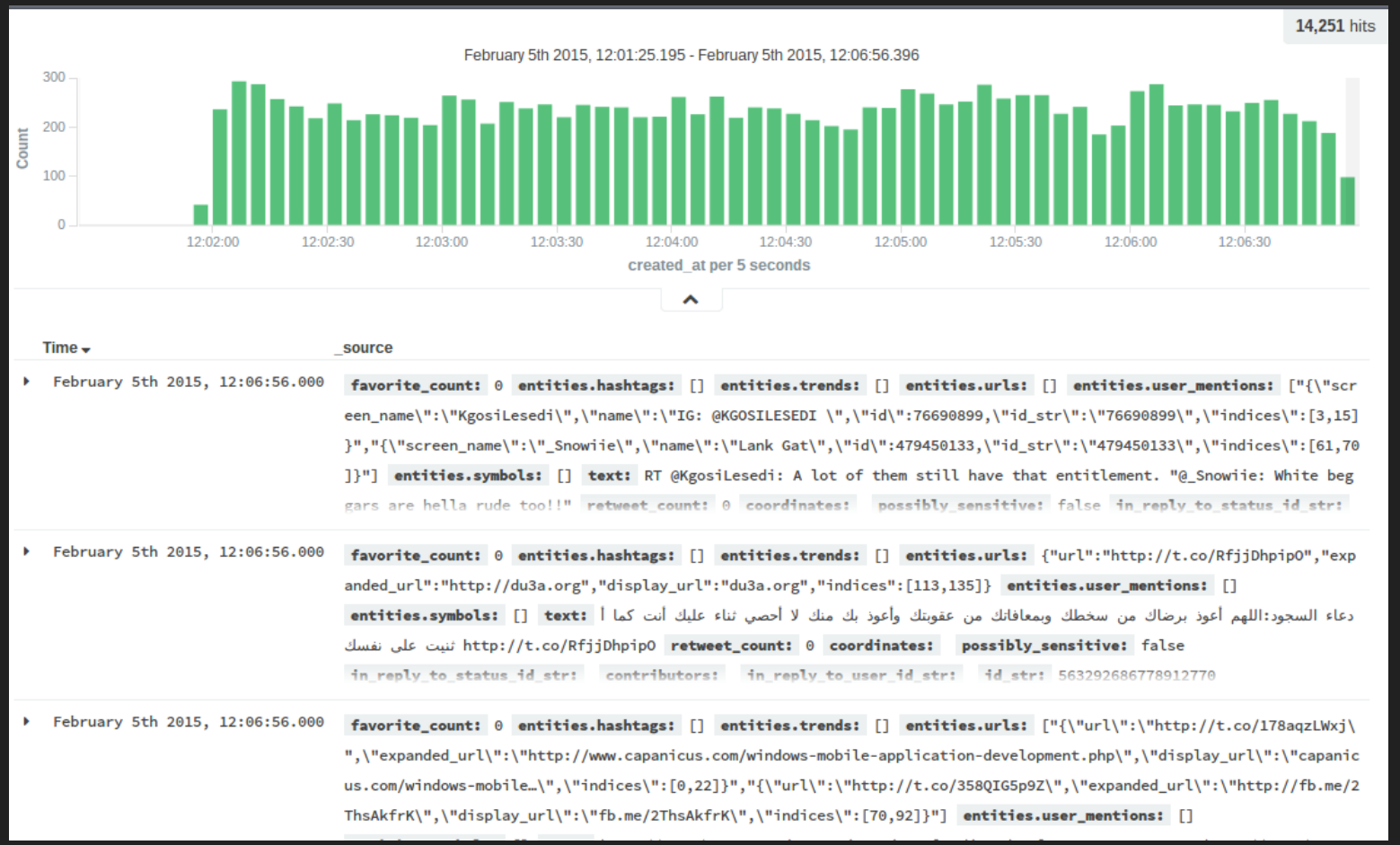
To this:



Kibana timestamp field

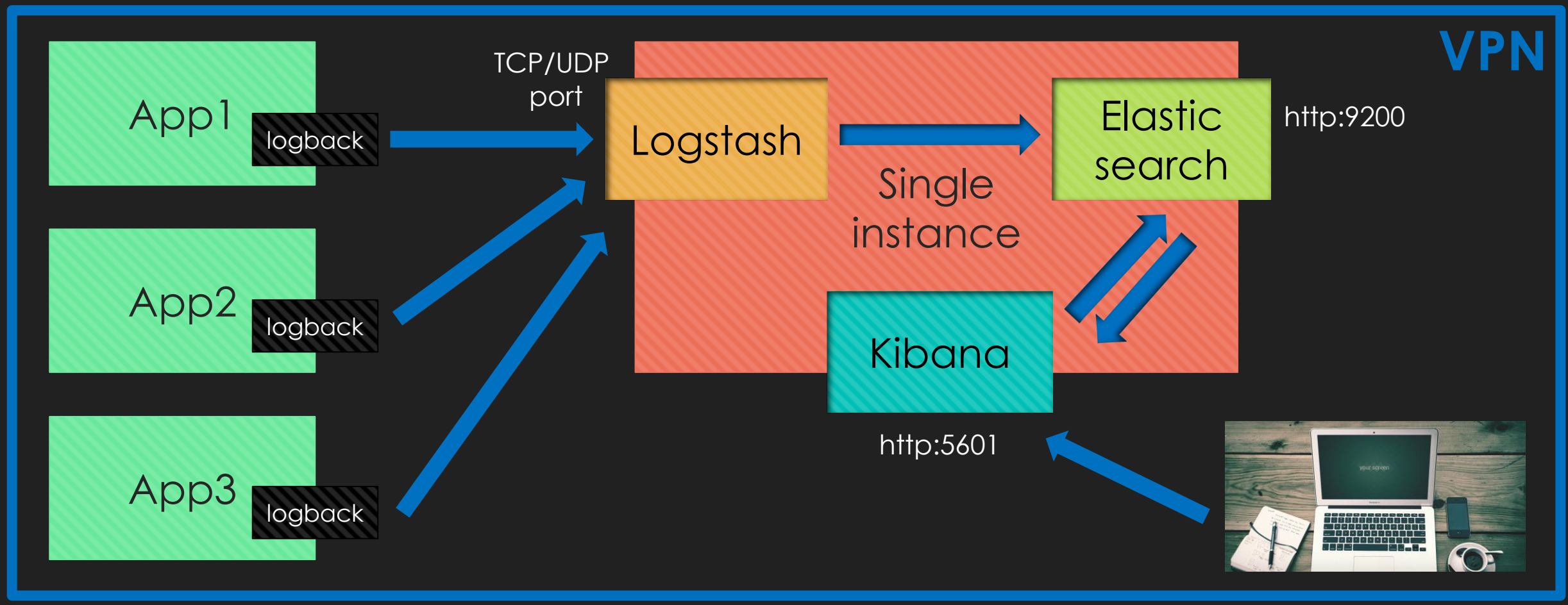
Not strings/objects

Events!

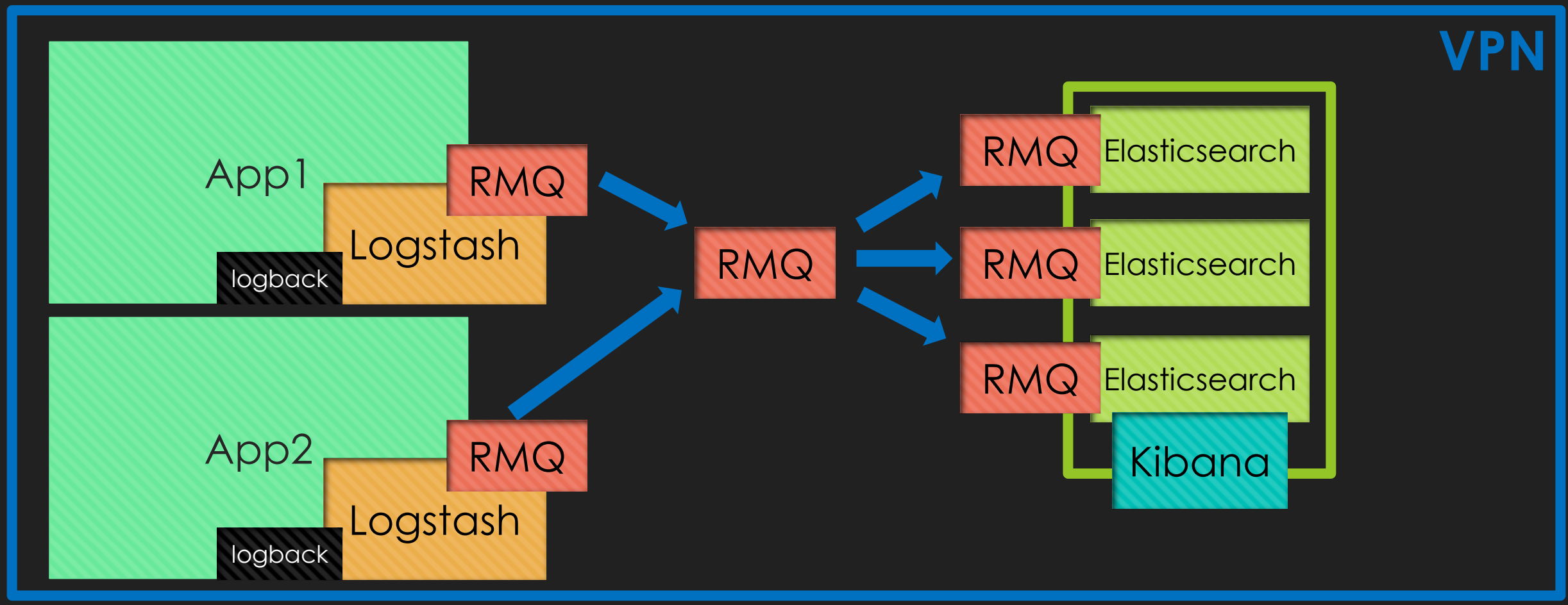


ELK Setups

Simple setup (startup mode)



Complex setup (QIWI)



Basic ELK usage scenarios

1. Search that whole cluster!

- Events from all your apps are in one place!
- Rocking useful and flexible timeline
- Powerful, text oriented search engine

2. Default logback fields

- Quickly filter events by:
  - Host name or IP
  - Log level, logger name
- Search only in stack trace.

3. Customize and save searches

- Save frequently used search
- Show only specific fields (e.g. message and host name)

Showtime

Advanced ELK.
Metrics and alerts.

1. Custom fields (search/filter)

Payment event:

```
{
  “user”     : “9112223344”,
  “amount”   : 500.00,
  “provider” : 464
}
```

Log custom fields (JVM)

```
libraryDependencies +=
   "net.logstash.logback" % "logstash-logback-encoder" % "4.5.1"
```

```
<appender name="LOGSTASH"
class="net.logstash.logback.appender.LogstashTcpSocketAppender">
  <destination>127.0.0.1:4560</destination>
</appender>

<root level="DEBUG">
  <!--...-->
  <appender-ref ref="LOGSTASH" />
</root>
```

Log custom fields (JVM)

```
import net.logstash.logback.marker.Markers._

val fields = Map[String, Any](
  "str_field" -> "str_value",
  "num_field" -> 666
)

logger.info(appendEntries(fields), "log message");
```

2. Elastalert

- Configured with a set of “rules”
- Performs periodical queries to ES
- Triggers alerts for matched rules

2. Elastalert rule types

- Frequency
- Spike
- Flatline
- Cardinality
- Blacklist, Whitelist
- Arbitrary query

```
# rules/500_frequency.yaml
es_host: localhost
es_port: 9200
name: 500 code frequency alert
type: frequency
index: logstash-*
num_events: 50
timeframe:
  minutes: 1
alert:
- "email"
email:
- “s.solonin@qiwi.ru"
```

2. Elastalert alert types

- Email
- Slack
- Jira
- Custom shell command

2. Elastalert with custom fields

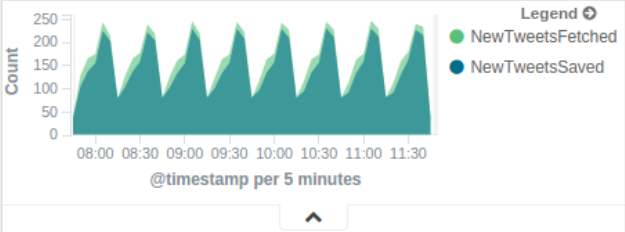
Any alert you can imagine can be implemented on custom fields:

- Average payment drops below 300 RUB
- Processing response latency spike
- User A buys something on Herbalife

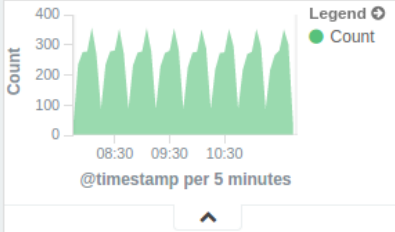
3. Monitoring and Metrics

- Realtime
- Lots of aggregations
- Beautiful visualizations
- Super power with custom fields

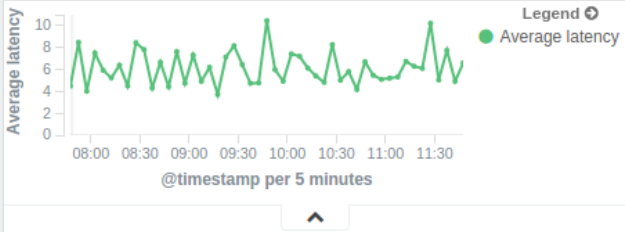
Fetched recent tweets [Graph]



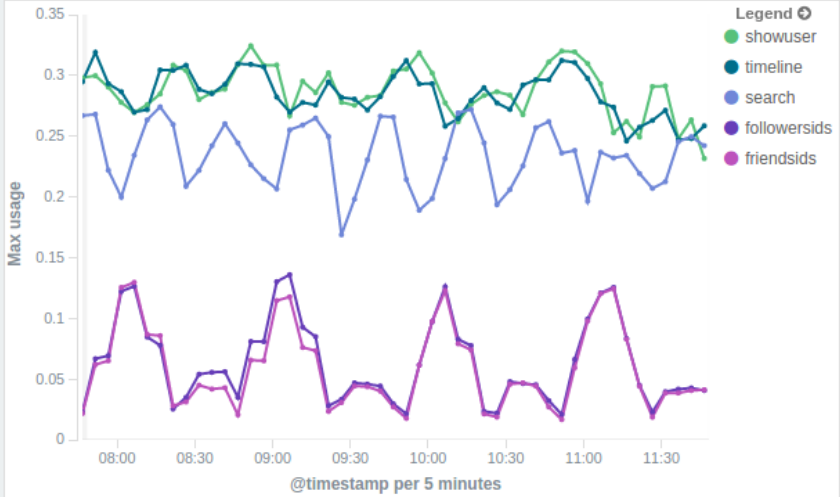
New ExpandedURL saved [Graph]



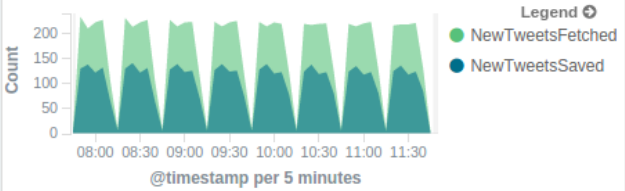
Twitter Utility Account Manager Actor Latency



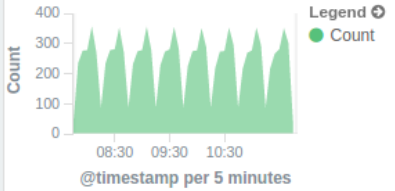
Utility Pool Exhauster



Fetched popular tweets [Graph]

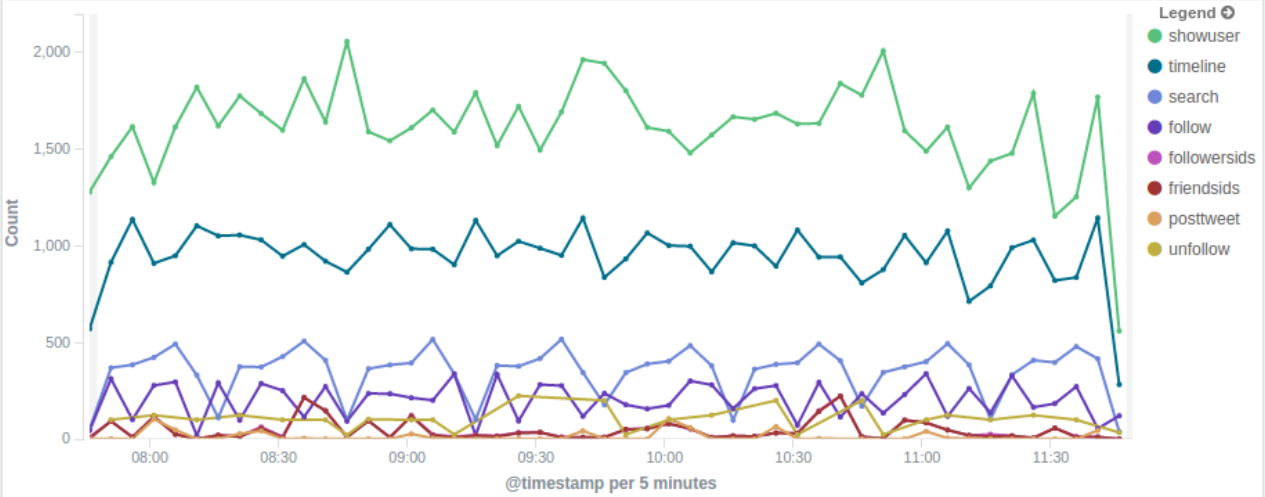


New ShortURL saved [Graph]

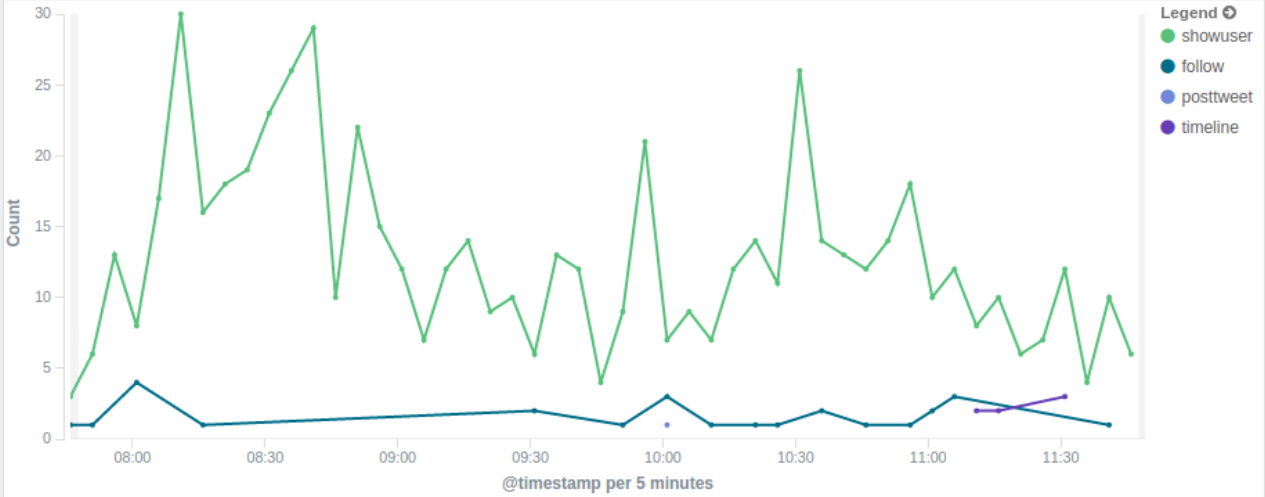


Number of follows (total)

Twitter API requests



Twitter API errors



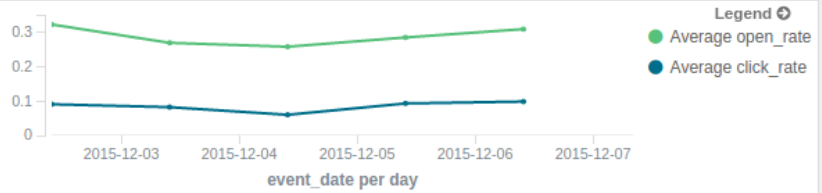
Total user...

671

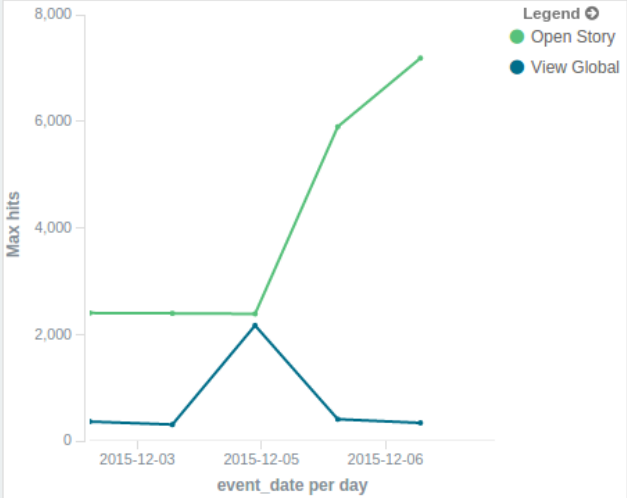
Max count

Total digest users and emails sent [Graph]

Digest open/click rates [Graph]



Mixpanel Events [Graph]

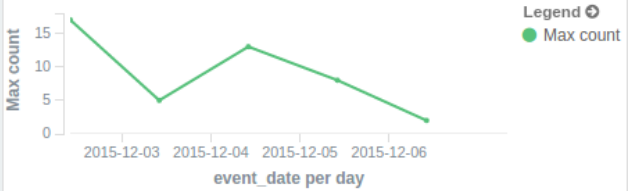


New user...

2

Max count

New digest users [Graph]



Yesterday open rate [Metric]

0.308

Average open_rate

Yesterday click rate [Metric]

0.098

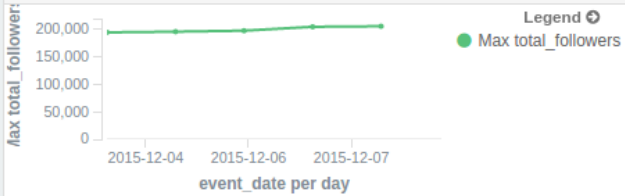
Average click_rate

Total followers [Metric]

205,977

Max total_followers

Total followers [Graph]

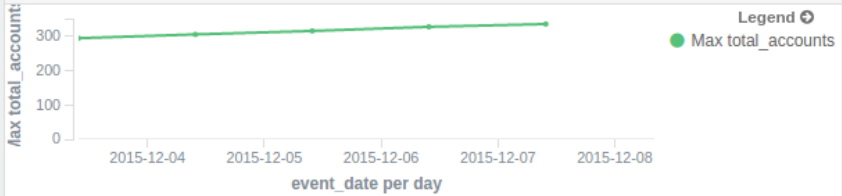


Total twitter accounts [Metric]

334

Max total_accounts

Total twitter accounts [Graph]

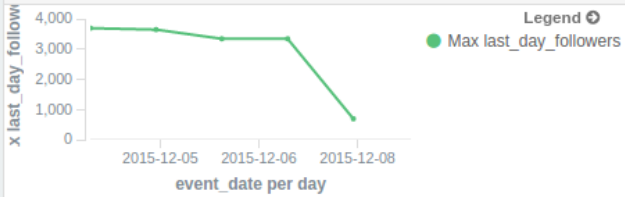


Last 24h followers [Metric]

692

Max last_day_followers

Recent followers [Graph]



Last 24h tweets [Metric]

2,781

Max total_tweets

Overall engagement [Metric]

0.205

Average total_engagement_score

Last 24h engagement [Metric]

0.257

Average last_day_engagement_score

Showtime

3. Why so useful?

- Tech monitoring (UI and flexibility)
- Business metrics (approx.)
- Alerts (both tech and business)
- Instant market/system reaction

Thanks !

kibana