ELK and Kafka

Tal levy

April 29, 2015

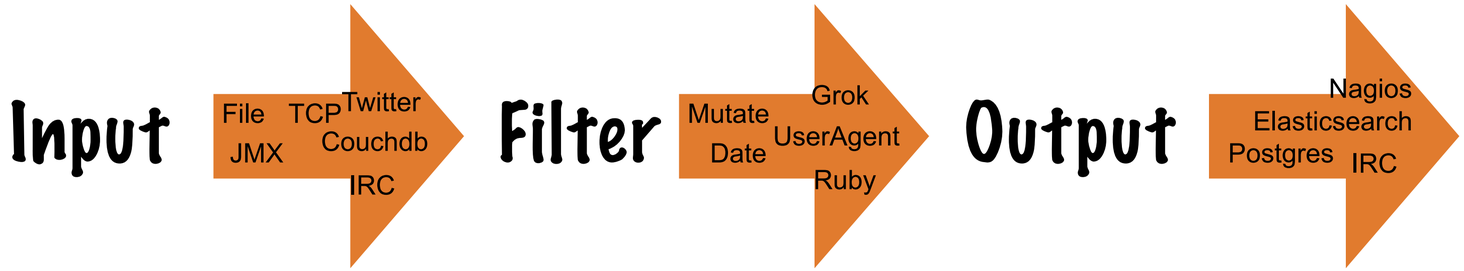
elastic

What

* Logstash
* Kafka and how Logstash can leverage it
* Message Serialization
* Monitoring Kafka with Elasticsearch, Logstash, Kibana

Logstash

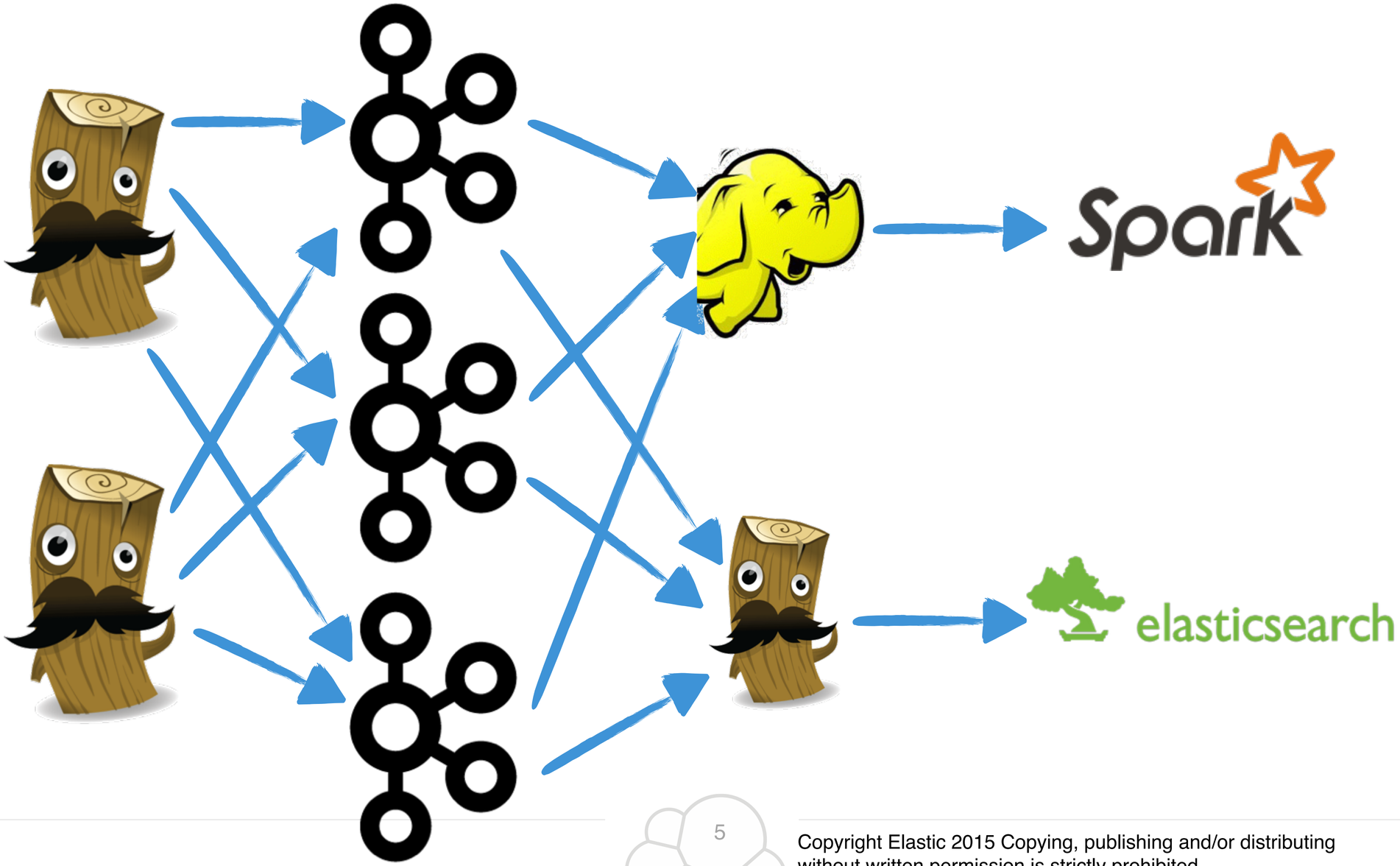
Log Processing Pipeline ...With LOTS of Plugins!



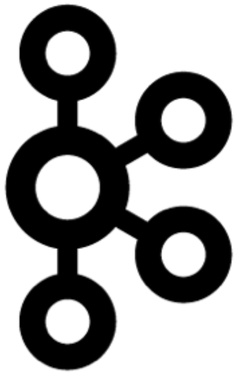
Logstash Config

```
input {
  file {
    path => "/path/to/logs"
  }
}

filter {
  grok {
     match => { "message" => "%{IP:client}" }
  }
}

output {
  elasticsearch {
    protocol => "http"
  }
}
```

Where



Kafka



* A High-throughput distributed messaging system

... and a Logstash plugin!

Input Plugin

```
input {
  kafka {
    topic_id => "most_important"
  }
}
```

Output Plugin

```
output {
  kafka {
    topic_id => "most_important"
  }
}
```

Custom Serializer

One of the options in Kafka Producer

```
output {
  kafka {
    ...
    serializer.class => "kafka.serializer.StringEncoder"
    ...
  }
}
```

Custom Dots Serializer

```
package org.logstash.dots;

import kafka.serializer.Encoder;
import kafka.utils.VerifiableProperties;

public class DotsEncoder implements Encoder<String> {
    public DotsEncoder(VerifiableProperties props) {
        //do nothing
    }

    @Override
    public byte[] toBytes(String s) {
        return ".".getBytes();
    }
}
```

https://github.com/talevy/kafka-dots-serializer

Custom Dots Serializer

```
output {
  kafka {
    ...
    serializer.class => "org.logstash.dots.DotsEncoder"
    ...
  }
}
```

Custom Serializer

In Logstash Environment

```
$ export CLASSPATH="$CLASSPATH:/path/to/serializer.jar"
$ bin/logstash -f /path/to/config
```

logs in, ... out!

Monitoring Kafka with ELK

elastic

Kafka & JMX

* Lots of metrics are exposed via JMX

... Logstash has a plugin for that!

JMX Plugin

jmx

edit

NOTE

This is a community-maintained plugin! It does not ship with Logstash by default, but it is easy to install by running `bin/plugin install logstash-input-jmx`.

Permits to retrieve metrics from jmx.

Synopsis

edit

This plugin supports the following configuration options:

Required configuration options:

```
jmx {
    path => ...
}
```

http://www.elastic.co/guide/en/logstash/current/plugins-inputs-jmx.html

Monitoring Server Throughput

jmx json configuration

```
{
"host": "localhost",
"port": 9999,
"alias": "kafka.server",
"queries": [
 {
   "object_name": "kafka.server:type=BrokerTopicMetrics,name=MessagesInPerSec,topic=*",
   "attributes": ["MeanRate"],
   "object_alias": "msg_per_sec-${topic}"
 }
]
}
```

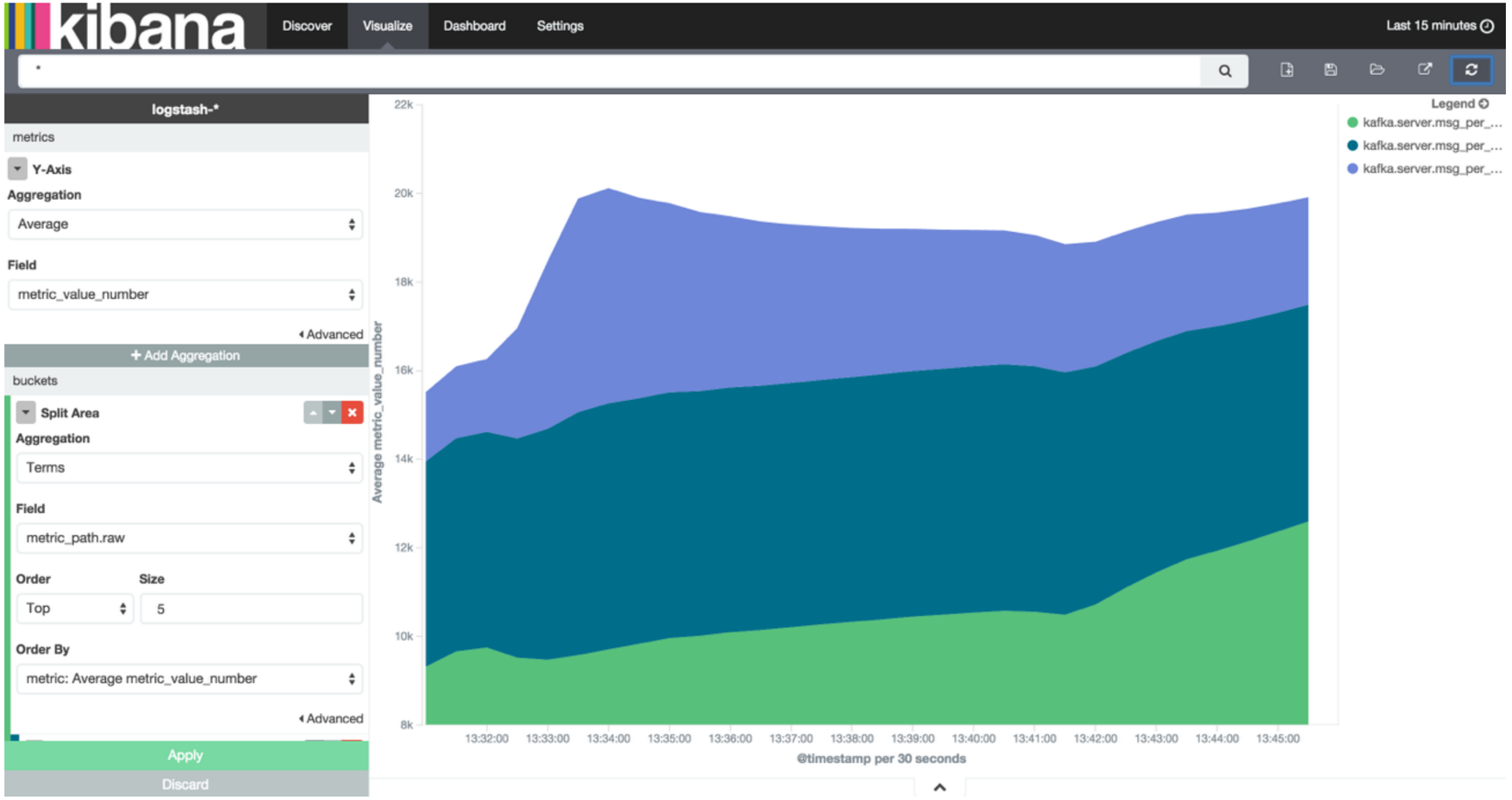
Logstash Config

```
input {
  jmx {
    path => "/path/to/jmxconf"
    polling_frequency => 5
  }
}

output {
  elasticsearch { protocol => "http" }
}
```

JMX Event

```
{
 "@version" => "1",
 "@timestamp" => "2015-04-29T21:59:10.160Z",
 "host" => "localhost",
 "path" => "jmx-server",
 "type" => nil,
 "metric_path" => "kafka.server.msg_per_sec-topic2.MeanRate",
 "metric_value_number" => 1982.548103009822
}
```

Throughput



Elasticsearch Index Management

Important to know if you want to save disk space!

* Daily Indices

  logstash-%{+YYYY.MM.dd}

* Curator

  A Python client for managing ES indices

Monitoring Consumption

custom java opts for Logstash

```
export LS_JAVA_OPTS="
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.port=3000
-Dcom.sun.management.jmxremote.ssl=false"
```

running Logstash kafka consumer:

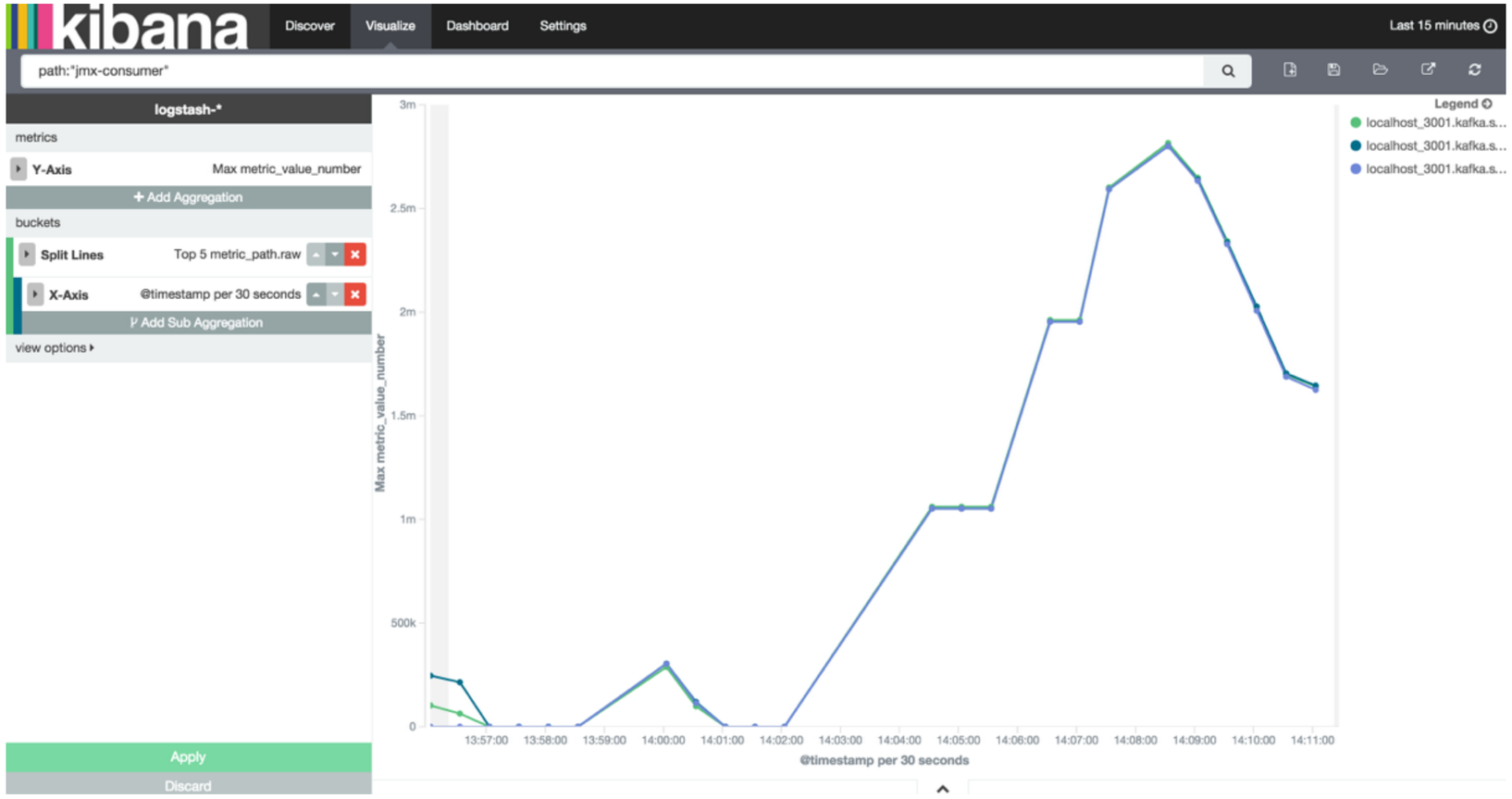
```
$ bin/logstash -e "input { kafka { topic_id => hello } }"
```

Monitoring Consumer Lag

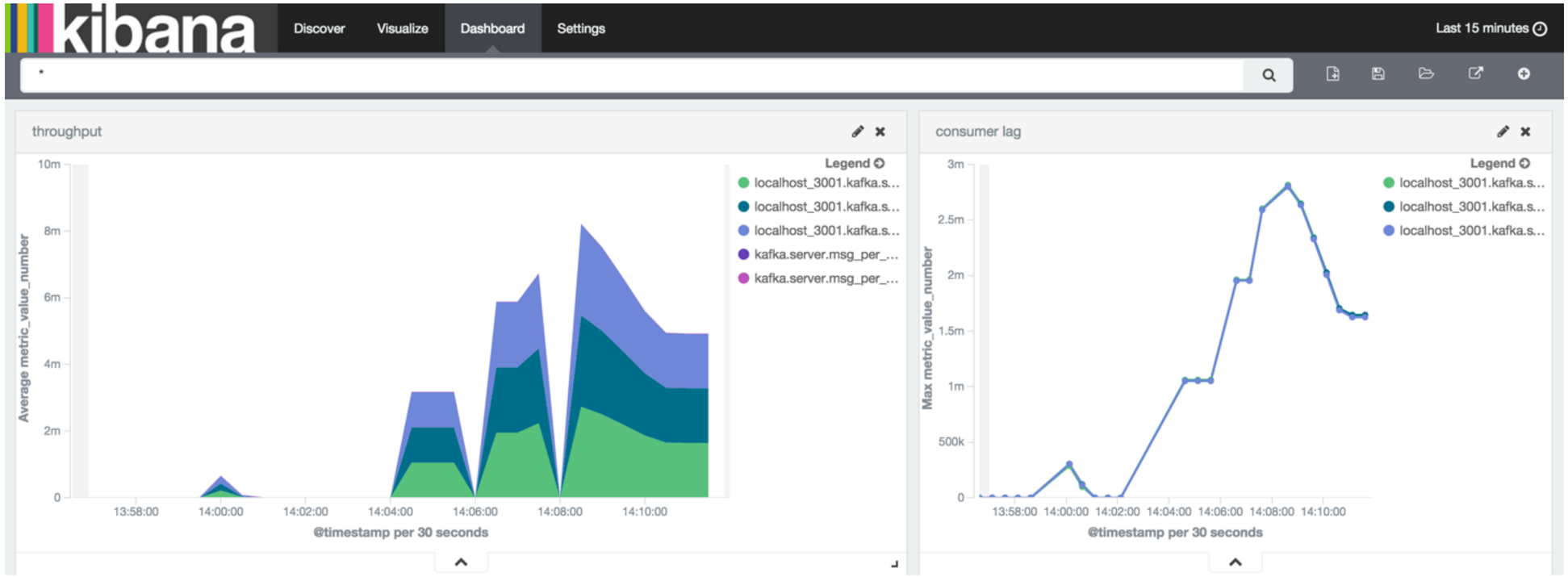
jmx json configuration

```
{
"host": "localhost",
"port": 3001,
"queries": [
  { "object_name": "kafka.server:type=FetcherLagMetrics,name=ConsumerLag,clientId=logstash,topic=*,partition=*" }
]
}
```

Consumer Lag



Together



Thank You!

elastic