Processing and visualising Hong Kong government open data with Elasticsearch, Kibana and Timelion

Joshua Rich

We’re gonna need some things…

Prerequisites

- Java 7 or 8 (OpenJDK FTW!)
- 20% free disk space
- Your operating system’s version of:
  - Elasticsearch 2.3.3
  - Logstash 2.3.2
  - Kibana 4.5.1
  - Download at https://elastic.co/downloads
- Datasets and configuration files:
  - Download at https://github.com/elastic/hkoscon16


OPEN DATA

Open data is releasing data

You probably have too much of it.

You probably can’t analyze it all yourself.

Everybody wins when data is free

Wider community/academic scrutiny.

Novel and thinking-outside-the-box use cases.

Artistic and personal interests.

Using Open Data

The 4 steps to success…

1. Find some cool data or data source you want to use.
2. Download all the data
3. …
4. Profit!

Using Open Data

What you actually need to do in Step 3...

1. Clean up data, including removing errors, reformatting to usable format
2. Set up data store for storing data
3. Create processing pipeline and process data into data store
4. Find you missed some errors
5. Clean up data some more
6. Repeat steps 1-3
7. Repeat steps 4-6
8. Reassess your life
9. Set up analysis pipeline and tooling
10. Analyse data, create visualisations and summaries

Elasticsearch
Logstash
Kibana

elasticsearch

Distributed & Scalable

- Resilient; designed for scale-out
- High availability; multitenancy
- Structured & unstructured data

Developer Friendly

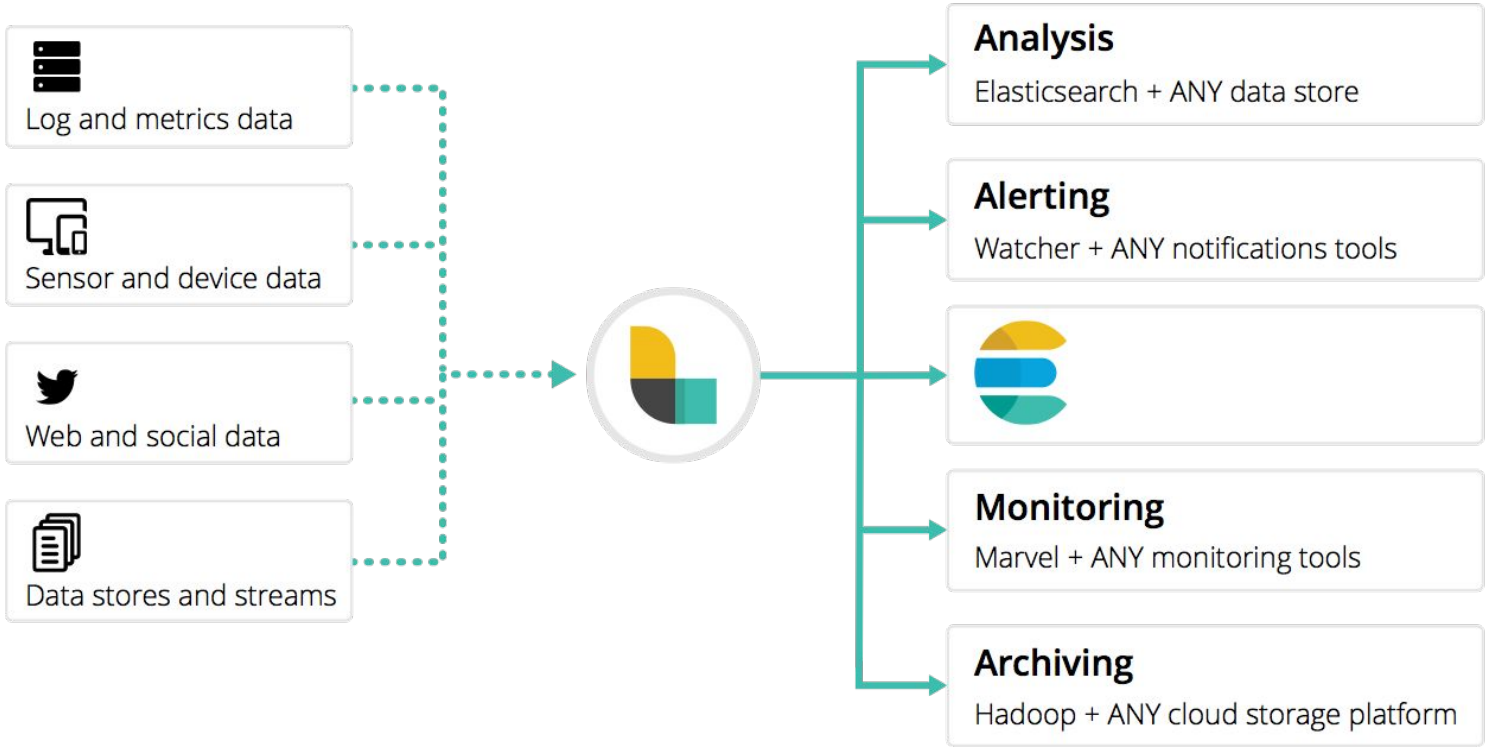
- Schemaless
- Native JSON
- Client libraries
- Apache Lucene

Search & Analytics

- Real-time
- Full-text search
- Aggregations
- Geospatial
- Multilingual

Collect, Enrich, and Transport

logstash



elastic

kibana

Discover Insights

- Explore and analyze patterns in data; drill down to any level
- Leverage powerful analytical capabilities in Elasticsearch

Customize & Share

- Create bar charts, line and scatter plots, maps and histograms
- Share and embed dashboards into operational workflows

Window into Elastic Stack

- Unified user interface for data visualization
- Administration and management for the Elastic Stack
- Pluggable architecture to create custom visualizations and applications

elastic

Hands-on Time!

Starting Elasticsearch

Open a new terminal window and type:

```
$ tar xvf elasticsearch-2.3.3.tar.gz # or unzip …
$ cd elasticsearch-2.3.3
$ bin/elasticsearch
[2016-06-09 15:51:29,541][INFO ][node                     ] [Devastator] version[2.3.3],
pid[22885], build[218bdf1/2016-05-17T15:40:04Z]
[2016-06-09 15:51:29,545][INFO ][node                     ] [Devastator] initializing
...
[2016-06-09 15:51:30,503][INFO ][plugins                  ] [Devastator] modules
[reindex, lang-expression, lang-groovy], plugins [], sites []
[2016-06-09 15:51:30,587][INFO ][env                      ] [Devastator] using [1] data
paths, mounts [[/ (/dev/mapper/fedora_josh--xps13-root)]], net usable_space [100.2gb],
net total_space [233.9gb], spins? [no], types [ext4]
[2016-06-09 15:51:30,588][INFO ][env                      ] [Devastator] heap size
[990.7mb], compressed ordinary object pointers [true]
[2016-06-09 15:51:34,724][INFO ][node                     ] [Devastator] initialized
[2016-06-09 15:51:34,724][INFO ][node                     ] [Devastator] starting ...
```

Starting Logstash

Open a new terminal window and type:

```
$ tar xvf logstash-2.3.2.tar.gz # or unzip …
$ cd logstash-2.3.3

# Run with a basic config specified on the command-line:

$ bin/logstash bin/logstash -e 'input { stdin { } } \
     output { stdout { codec => "rubydebug" } }'
Settings: Default pipeline workers: 4
Pipeline main started

# Now type something in your terminal...
```

Starting Kibana

Open a new terminal window and type:

```
$ tar xvf kibana-4.5.1-linux-x64.tar.gz # package name will vary depending on OS...
$ cd kibana-4.5.1

# Install timelion, we will use this later...

$ bin/kibana plugin --install elastic/timelion

# Install sense, we will use this later...

$ bin/kibana plugin --install elastic/sense

# Start Kibana

$ bin/kibana
```

The Datasets

Hong Kong Air Pollution Index

- Calculated from several individual pollutant measurements.
- In use between June 1995 and December 2013.
- Gathered from 14 stations around Hong Kong, three of which are located roadside.
- Provides overall idea of air quality in easily understandable metric.
- Our data covers 2000 - 2013

API ratings and health effects

API	Level	Guidelines
0-25	Low	Enjoy outside activities as normal
26-50	Medium	Enjoy outside activities as normal
51-100	High	Acute health effects are not expected but chronic effects may be observed if one is persistently exposed to such levels.
101-200	Very High	People with existing heart or respiratory illnesses may notice mild aggravation of their health conditions. Generally healthy individuals may also notice some discomfort.
201-500	Severe	People with existing heart or respiratory illnesses may experience significant aggravation of their symptoms. There may also be widespread symptoms in the healthy population (e.g. eye irritation, wheezing, coughing, phlegm and sore throats).

Hong Kong Daily Weather Observations

- Historical daily measurements:
- Temperatures, humidity, pressure, rainfall, wind, etc.
- Original data spanned 1997-2015, we’re using a subset to align with the API data.

Indexing the data

elasticsearch

logstash

Logstash and Data

Three stage processing pipeline:

Listen for data

Poll for data

Interpreting formats
(json, xml, csv, text)

Normalise values
(timestamp formats)

Substitute values
(lookup tables)

Remove/Add values

Correct inconsistencies

Drop junk events

Secondary Processing
(send to another Logstash)

Buffering
(send to a broker)

Indexing
(send to Elasticsearch)

Logstash - Input

Read csv file

```
input {
    file {
        path => "hongkong-weather.csv"
        start_position => "beginning"
        type => "weather"
    }
}
```

Logstash - Filter Part 1

```
filter {
    # don’t read the csv file header
    if [message] !~ /^\d+/ {
        drop {}
    }
    # assign names to the csv file columns
    csv {
        columns => ["date","average_api","max_api","min_api","station_name","
station_type","latitude","longitude"]
        remove_field => ["message"]
    }
...
```

Logstash - Filter Part 2

```
...
    # Add a new combined lat/long field called “location”
    mutate {
        add_field => { "location" => "%{latitude},%{longitude}" }
    }
    # Properly assign the date to the event
    date {
        match => ["date", "YYYY-MM-dd", "YYYY/MM/dd", "dd-MM-YYYY", "dd/MM/YYYY"]
        remove_field => "date"
    }
}
```

Logstash - Output

```
output {
    # Index into the “hk-api” index
    elasticsearch {
        index => "hk-api"
    }
    # watch the dots...
    stdout {
        codec => "dots"
    }
}
```

Let’s do it!

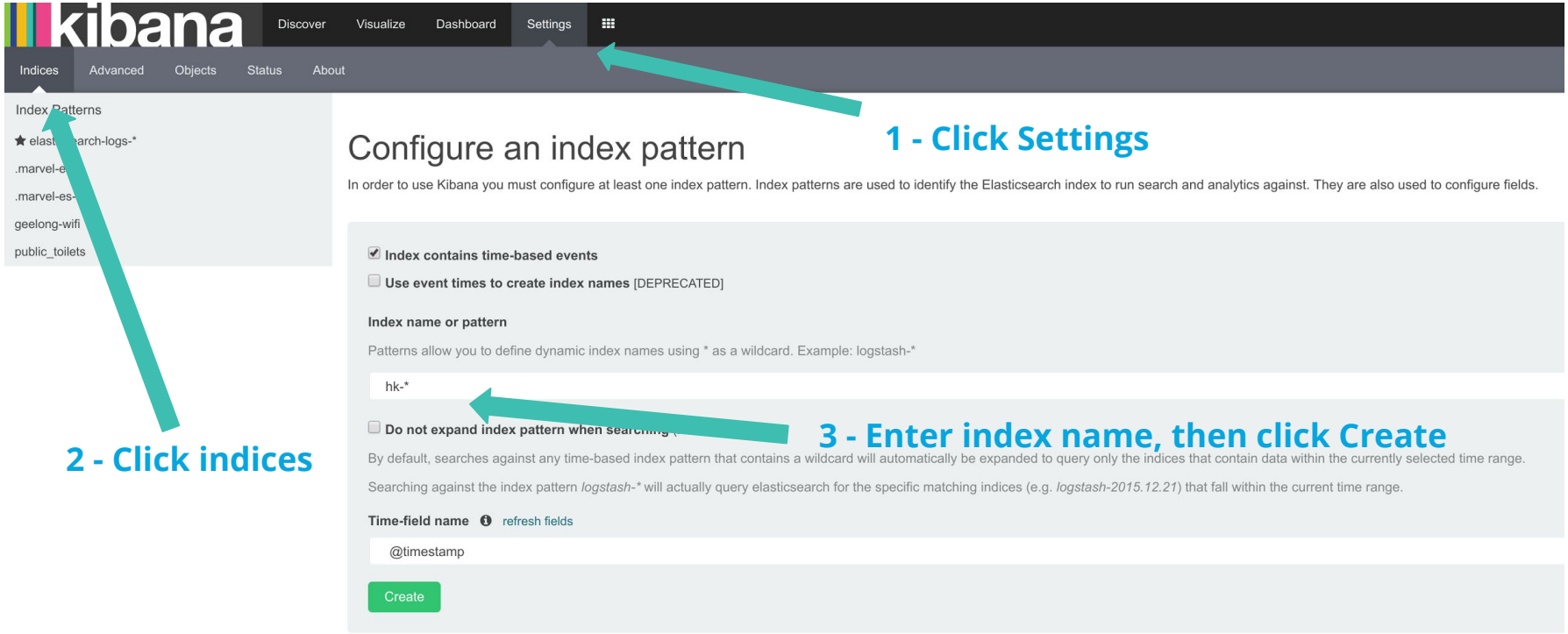
Open a new terminal window and type:

```
# change into the api data directory:
$ cd /path/to/data/api
# pipe the csv files to Logstash using the config:
$ cat *csv | /path/to/logstash-2.3.1/bin/logstash -f ./hongkong-api-
logstash.conf
# watch the dots...

# change into the weather data directory:
$ cd /path/to/data/weather
# pipe the csv files to Logstash using the config:
$ cat *csv | /path/to/logstash-2.3.1/bin/logstash -f ./hongkong-
weather-logstash.conf
# watch the dots...
```

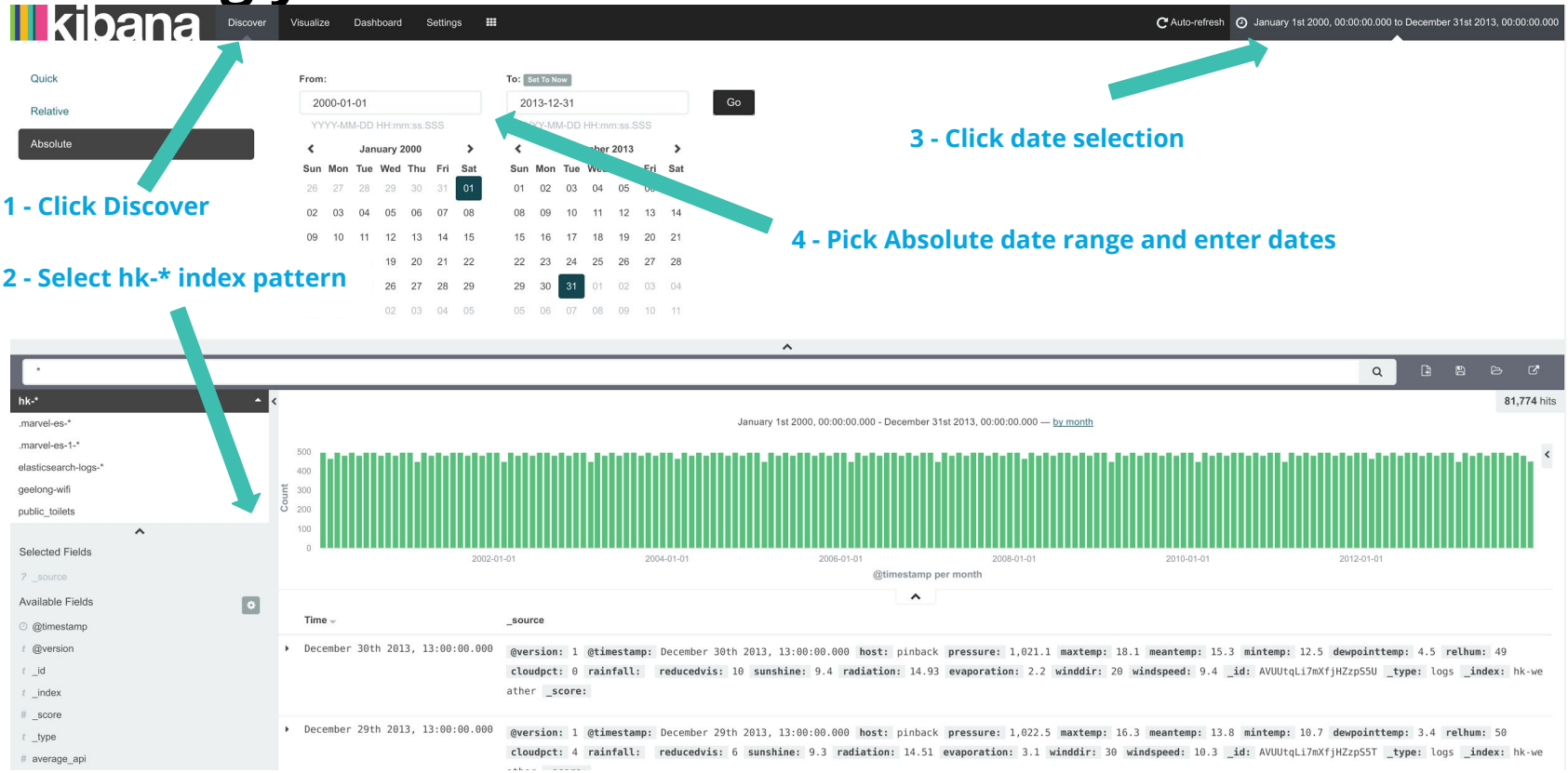
Analysing and Visualising the data

Create an index pattern



elastic

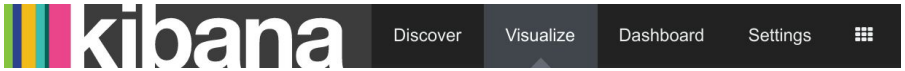
Browsing your data with Discover



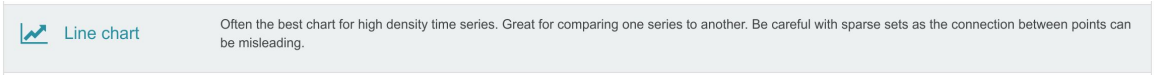
elastic

Creating your first visualisation

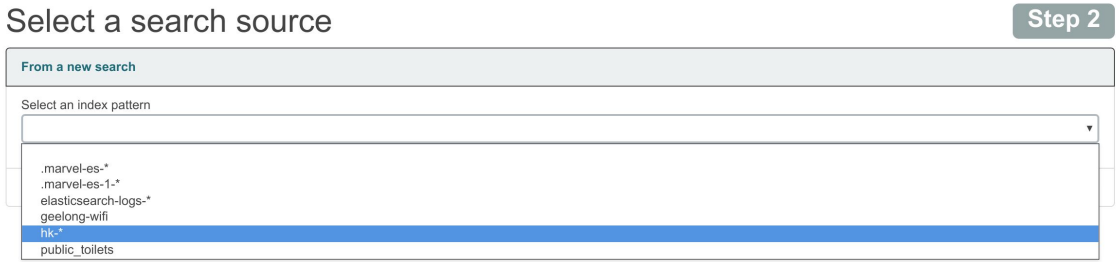
1. Click **Visualize**:



2. Pick a **Line Chart**:



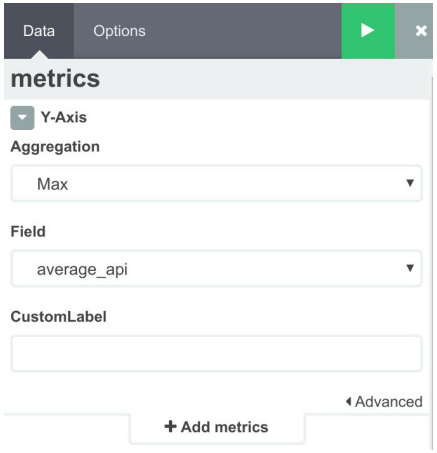
3. Choose **From a new search** and pick the **hk-*** pattern:



elastic

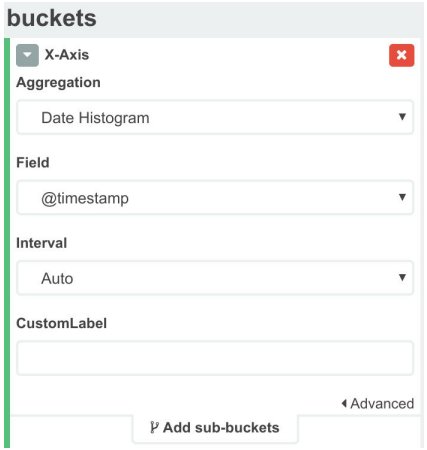
For **metrics/Y-Axis**:

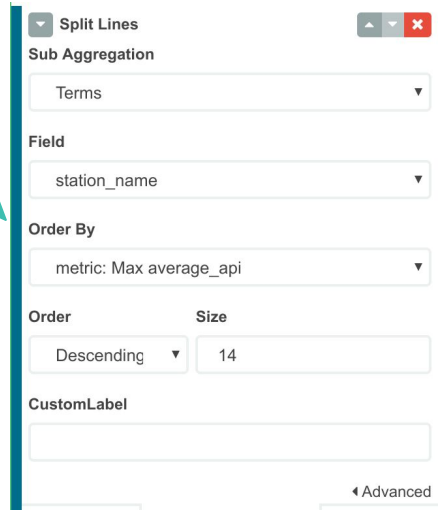
- Choose *Max* **Aggregation**
- Choose *average_api* **Field**



For **buckets/X-Axis**:

- Choose *Date Histogram* **Aggregation**
- Leave other options as defaults
- Click **Add sub-buckets**

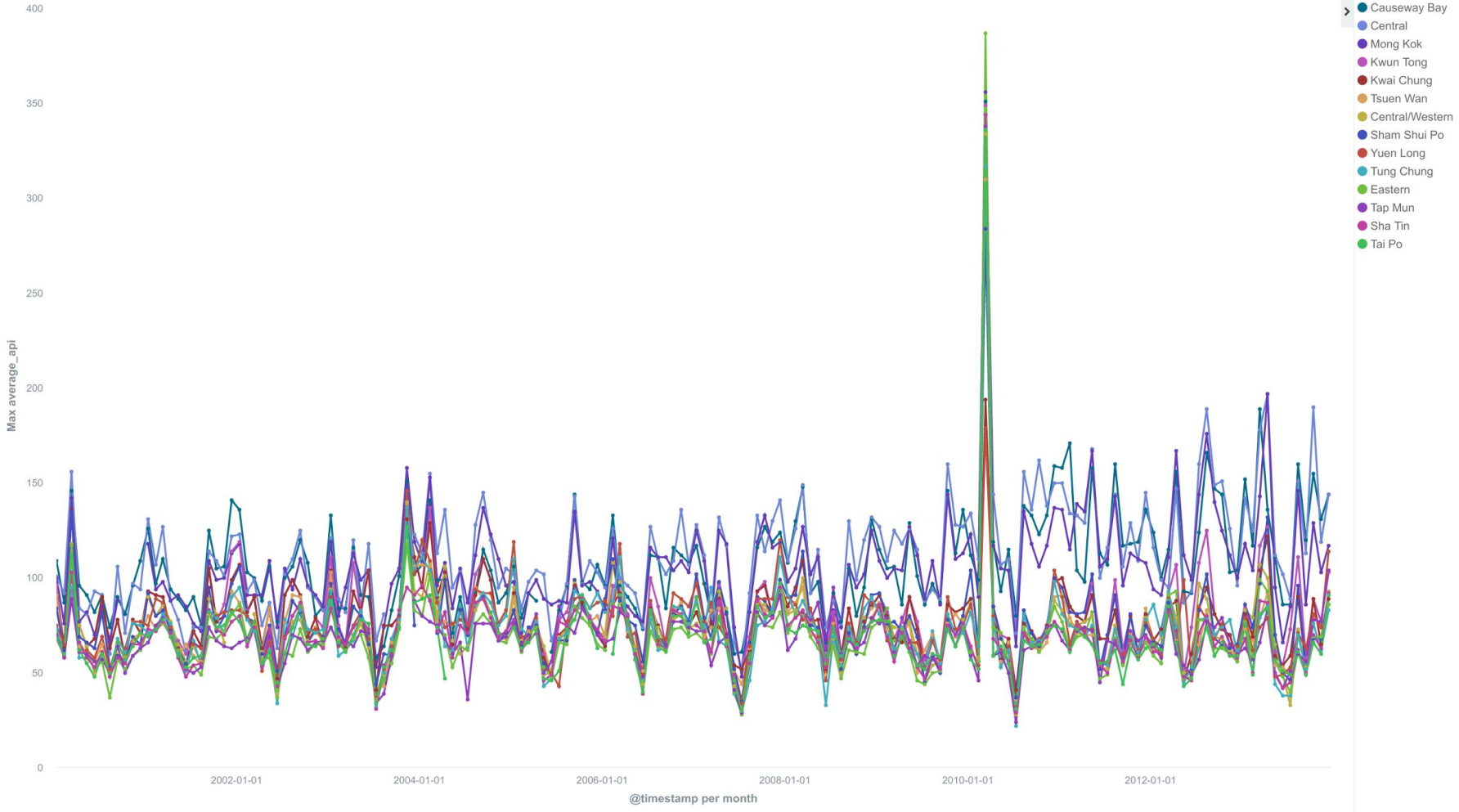




For **sub-aggregation**:

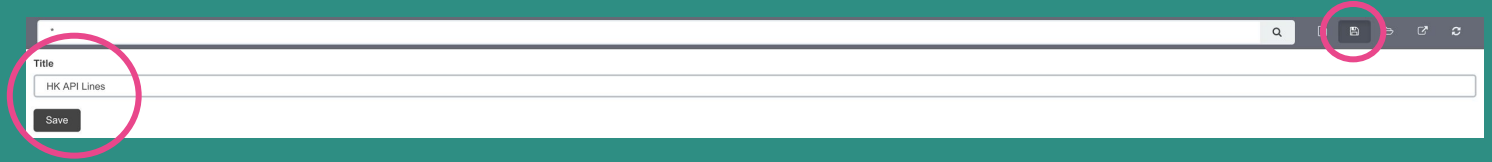
- Choose *Terms* **Aggregation**
- Choose *station_name* **Field**
- Set **Size** to *14*

elastic


Causeway Bay
Central
Mong Kok
Kwun Tong
Kwai Chung
Tsuen Wan
Central/Western
Sham Shui Po
Yuen Long
Tung Chung
Eastern
Tap Mun
Sha Tin
Tai Po
400
350
300
250
200
150
100
50
0
Max average_api
2002-01-01
2004-01-01
2006-01-01
2008-01-01
2010-01-01
2012-01-01
@timestamp per month

elastic

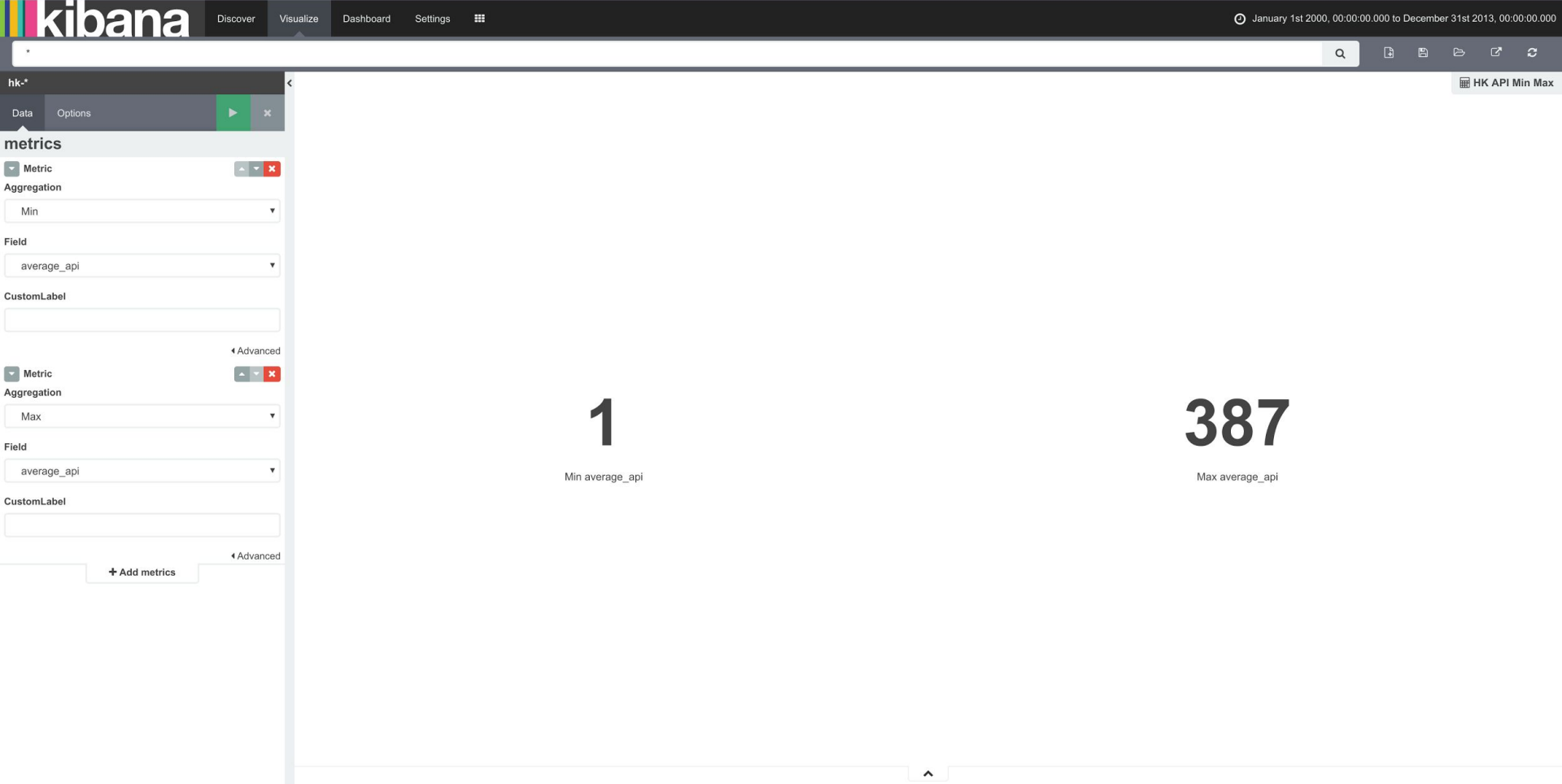
Save it!



elastic

Creating a second visualisation

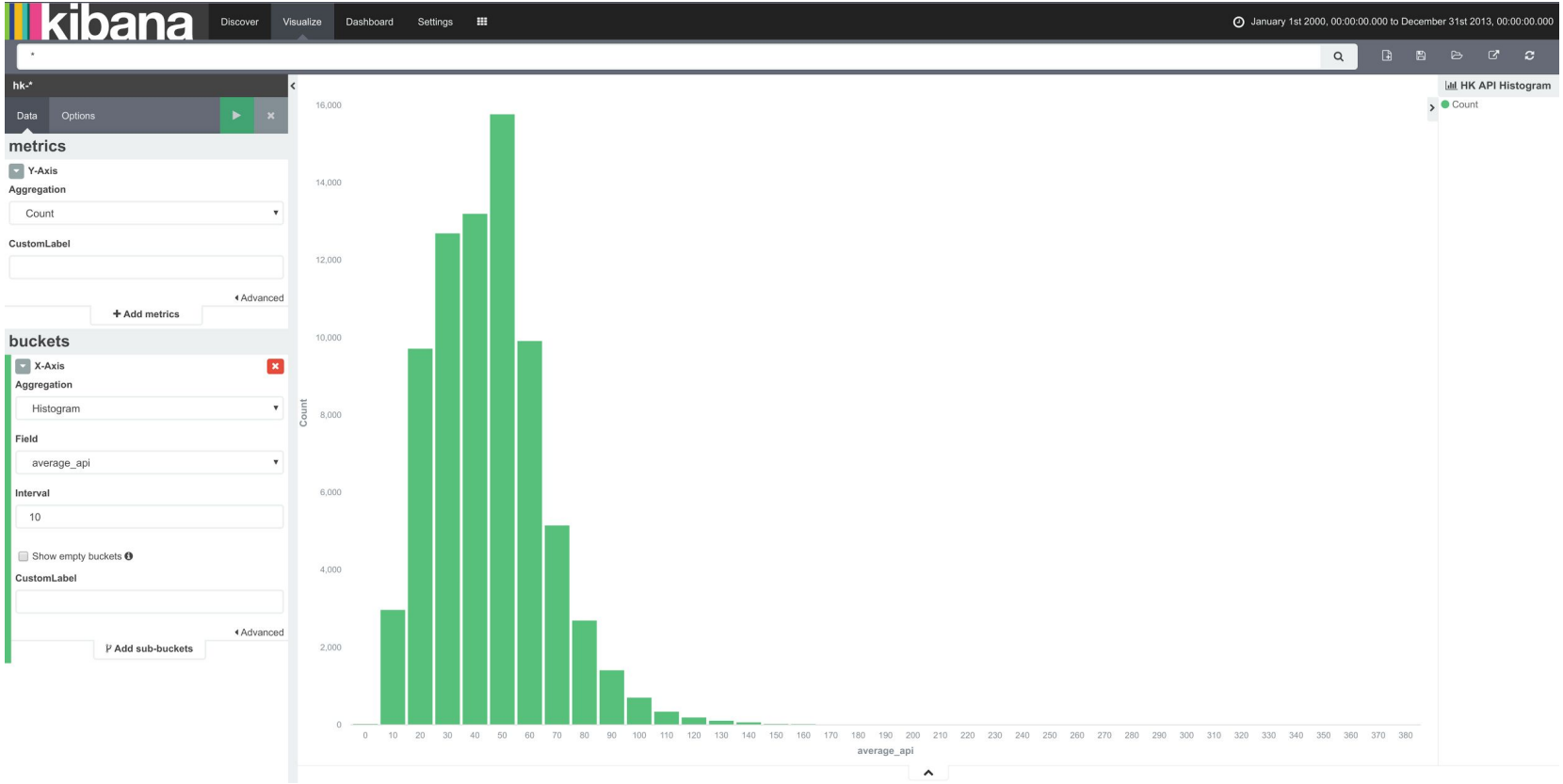
1. Click **Visualize**
2. Pick a **Metric**
3. Choose **From a new search** and pick the **hk-*** pattern
4. Click **Metric**, choose a *Min* **aggregation** on the **field** *average_api*
5. Repeat 4, but choose a *Max* **aggregation.**


kibana
Discover
Visualize
Dashboard
Settings
January 1st 2000, 00:00:00.000 to December 31st 2013, 00:00:00.000
*
hk-*
Data
Options
metrics
Metric
Aggregation
Min
Field
average_api
CustomLabel
Advanced
Metric
Aggregation
Max
Field
average_api
CustomLabel
Advanced
+ Add metrics
HK API Min Max
1
Min average_api
387
Max average_api

elastic

Save it again!

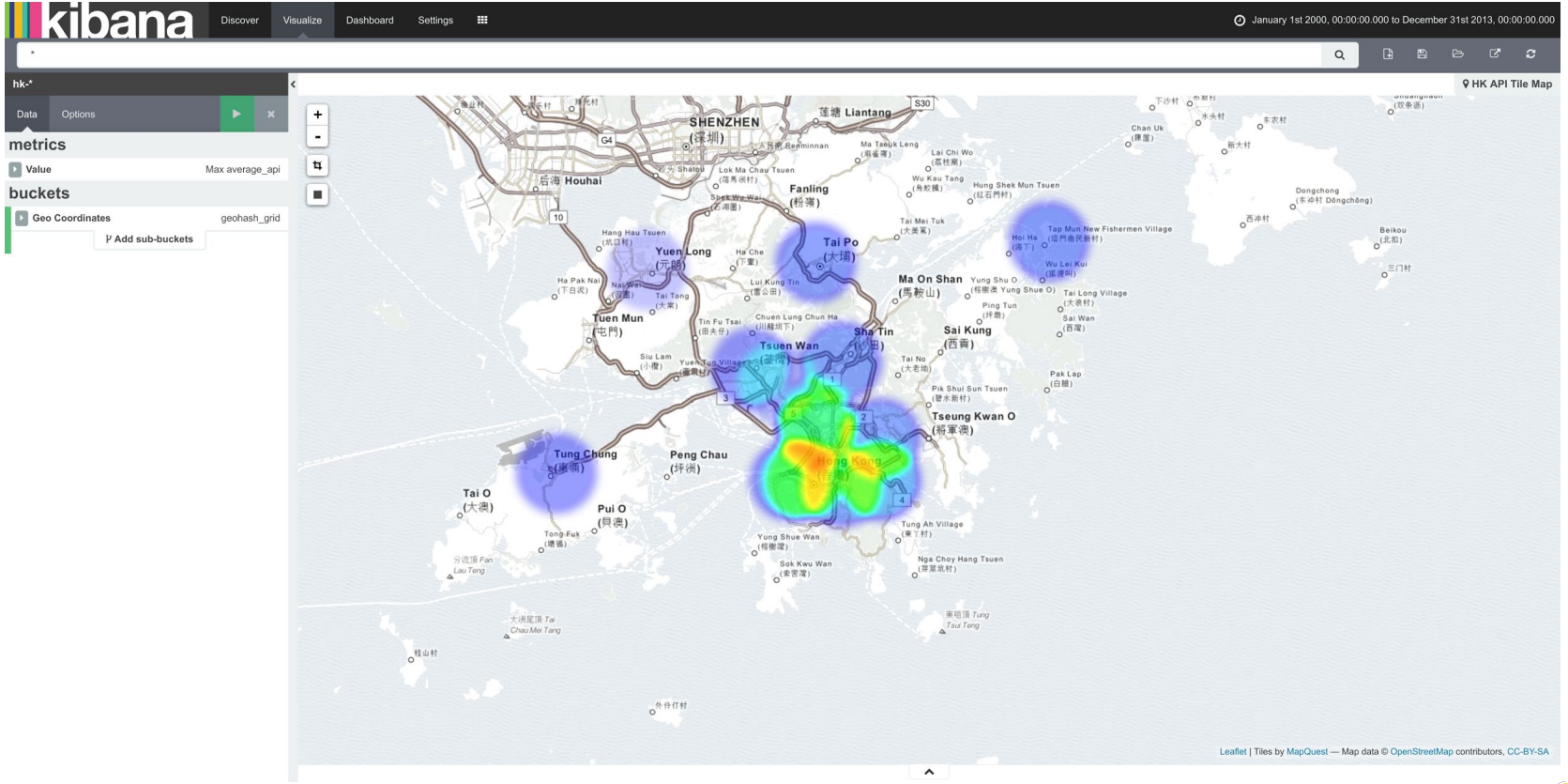
Now let’s create this…



elastic

Save it again!

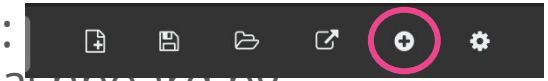
And this…

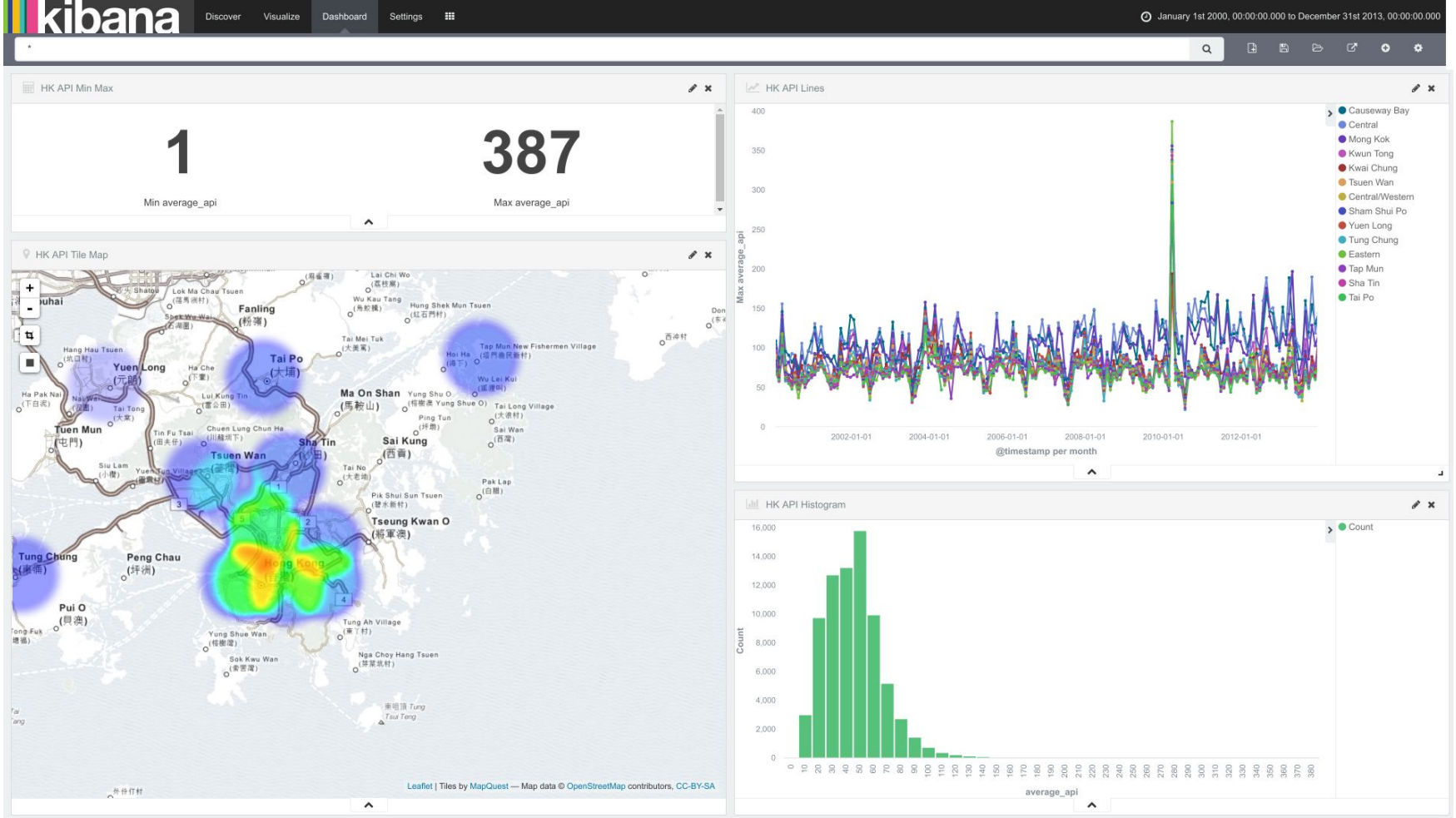


elastic

Save it again!

Putting Visualizations together in a Dashboard

1. Click **Dashboard**
2. Click the *plus* button to the right of the search bar: 
3. Add all the visualizations you just created to the dashboard by clicking their names.
4. Now adjust them to create this...


kibana
Discover
Visualize
Dashboard
Settings
January 1st 2000, 00:00:00.000 to December 31st 2013, 00:00:00.000
HK API Min Max
1
Min average_api
387
Max average_api
HK API Lines
Max average_api
@timestamp per month
Causeway Bay
Central
Mong Kok
Kwun Tong
Kwai Chung
Tsuen Wan
Central/Western
Sham Shui Po
Yuen Long
Tung Chung
Eastern
Tap Mun
Sha Tin
Tai Po
HK API Tile Map
Fanling
Tai Po
Yuen Long
Tuen Mun
Tsuen Wan
Sha Tin
Ma On Shan
Sai Kung
Tseung Kwan O
Tung Chung
Peng Chau
Pui O
Hong Kong
Leaflet | Tiles by MapQuest — Map data © OpenStreetMap contributors, CC-BY-SA
HK API Histogram
Count
average_api

elastic

Exploring the data

Powered by Elasticsearch queries

In the search bar, type **average_api:<25**

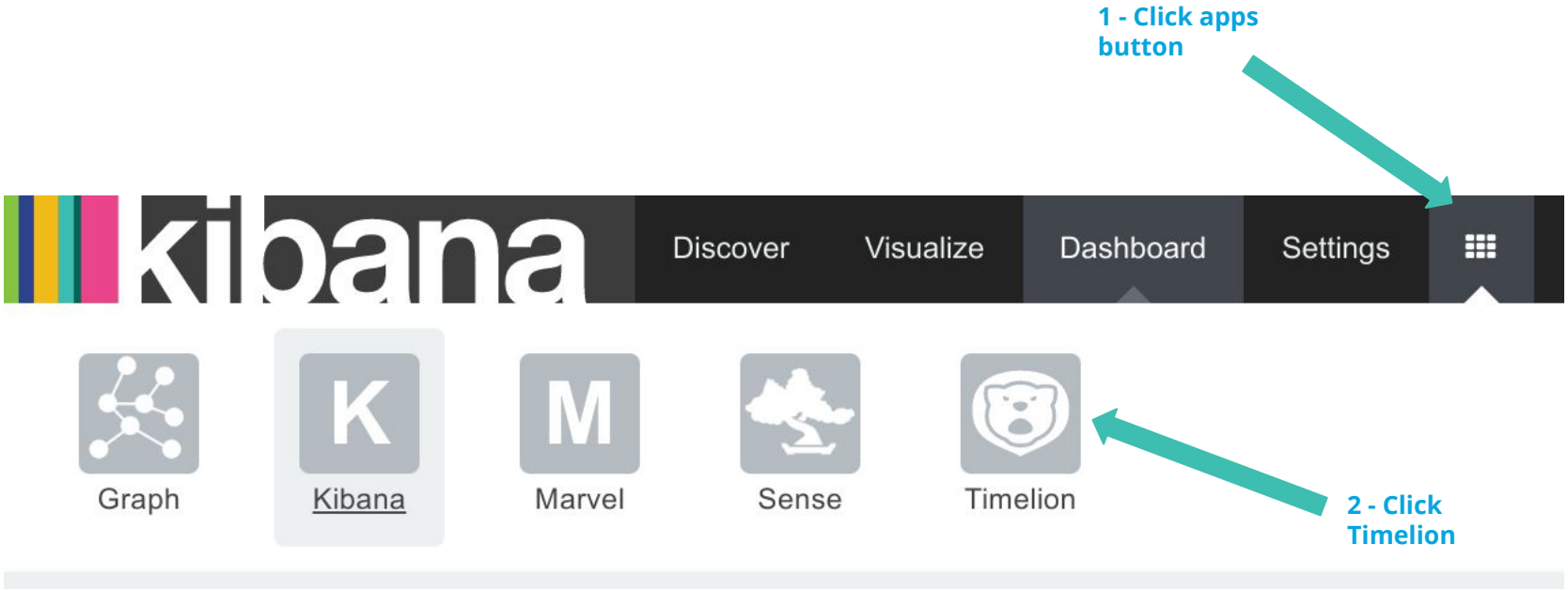
- **What happened?**

Try these searches:

- **station_name:”Kwun Tong”**
- **Station_type:roadside**

What else can you see by searching and exploring this data?

Going deeper on time-series with Timelion

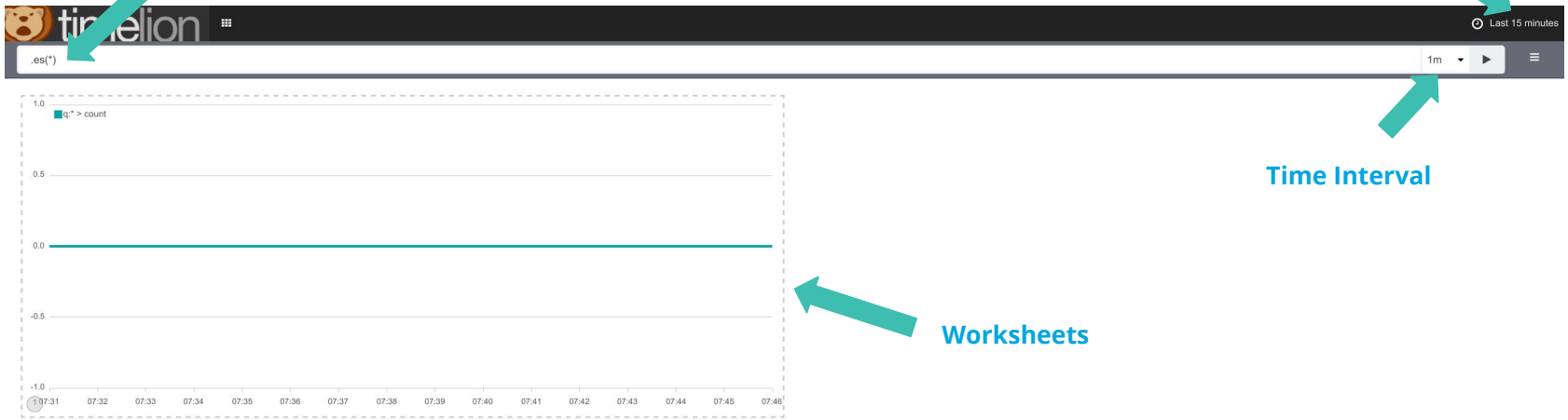


elastic

Getting around

Query Bar

Time Picker



Time Interval

Worksheets

elastic

Let’s take a look at the average API by station again…

Enter the following in the query bar:

```
.es(index=hk-*,metric=max:average_api,split=station_name:14)
```

What do you see?

Add the following to the end:

```
.mvavg(window=31)
```

What happened?

Change `split=station_name:14` to `split=station_type:2`

What do you see now?

Is pollution affected by rain?

Change the Time Interval to *1w*

Enter the following in the search bar:

```
.es(index=hk-*,metric=max:average_api).mvavg(31)  .es(index=hk-*,
metric=max:rainfall).mvavg(31)
```

What do you see?

Moving on to trending topics…

Try this query:

```
.es(index=hk-*,metric=max:maxtemp).mvavg(52)
```

Is there an overall trend in maximum temperature over the years?

What could cause this?

Wrapping it up

What have we achieved?

- We **install**ed a base ELK stack
- We **import**ed some csv data
- We **visualise**d the data in Kibana
- We **explore**d the data with some searching and filtering
- We **analyze**d the data to discover some trends and inferred details

With these powers combined...

- Elasticsearch can store your data and provide fast searching, analytics and retrieval capabilities
- Logstash can read your data, clean it, reformat it and write it to Elasticsearch
- Kibana can access Elasticsearch providing you a nice GUI to search and visualise your data in a variety of ways.

elastic

Competition Time!

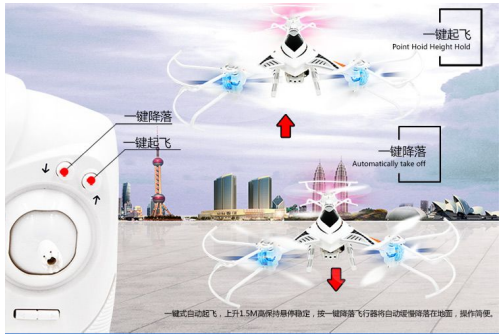
Twitter/Facebook Competition

#madewithelastic

- We want you to take what you've learned today and show us what is possible.
- Find some open data.
- Import it into Elasticsearch and create a Dashboard showing it off in Kibana.
- Mention **@elastic** in a tweet or post on our FB account wall with:
  - screenshot of your dashboard
  - a brief description
  - hashtag **#madewithelastic**
- The best/most interesting/coolest dashboard win prizes...

First Prize:

Smartphone controlled
quadcopter!
Six axis, WiFi controlled



elastic

Second and Third Prizes:

- A 1.2GHz 64-bit quad-core ARMv8 CPU
- 1GB RAM
- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot (now push-pull rather than push-push)
- VideoCore IV 3D graphics core
- 802.11n Wireless LAN
- Bluetooth 4.1, Bluetooth Low Energy (BLE)

elastic

Where to find open data

- Hong Kong Open Data:
  - https://data.gov.hk
- Awesome Public Datasets:
  - https://github.com/caesar0301/awesome-public-datasets
- Stack Exchange Data Dump:
  - https://archive.org/details/stackexchange
- Google Trends Datastore:
  - https://googletrends.github.io/data/

Thank you!

Twitter Competition **#madewithelastic** Details

- Find some open data.
- Import it into Elasticsearch and create a Dashboard showing it off in Kibana.
- Mention **@elastic** in a tweet or post on our FB account wall with:
  - screenshot of your dashboard
  - a brief description
  - hashtag **#madewithelastic**
- The best/most interesting/coolest dashboards win prizes!