A Kibana Tutorial: Getting Started

By: Daniel Berman

#Note: Elastic recently announced it would implement closed-source licensing for new versions of Elasticsearch and Kibana beyond Version 7.9. For more details, read our CEO Tomer Levy's comments on Truly Doubling Down on Open Source.

Kibana is the visualization layer of the ELK Stack—the world's most popular log analysis platform which is comprised of Elasticsearch, Logstash, and Kibana.

More on the subject:

- Configuring YAML Files after Installing the ELK Stack
- Logging Kubernetes on GKE with the ELK Stack and Logz.io
- The Importance of Security in IoT

This tutorial will guide you through some of the basic steps for getting started with Kibana—installing Kibana, defining your first index pattern, and running some basic searches. We'll also compare Kibana with OpenSearch Dashboards, the new forked version of Kibana launched by AWS after Elastic decided to close source Elasticsearch and Kibana in early 2021.

This tutorial is for anyone curious to install Kibana on their own. It assumes you've already got the database (Elasticsearch) and parsing tool (Logstash) configured and ready to go.

Need help installing Elasticsearch? Check out this Elasticsearch tutorial. If you haven't installed Logstash yet, or are not familiar with how to use it, check out this Logstash tutorial.

If you'd rather not spend the time or resources installing and managing your own ELK Stack, Logz.io Log Management offers a fully managed service based on OpenSearch and OpenSearch Dashboards, which provides log ingestion, parsing, storage, and analysis dashboards out-of-the-box, at any scale.

Installing Kibana

Presuming you already have Elasticsearch installed and configured, we will start with installing Kibana. If you want to find out more about installing Elasticsearch, check out this Elasticsearch tutorial.

Depending on your operating system and your environment, there are various ways of installing Kibana. We will be installing Kibana on an Ubuntu 16.04 machine running on AWS EC2 on which Elasticsearch and Logstash are already installed.
Start by downloading and installing the Elastic public signing key:

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |
sudo apt-key
add -
```
Copy
Add the repository definition:

```
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable
main" | sudo
tee -a /etc/apt/sources.list.d/elastic-7.x.list
```
Copy
It's worth noting that there is another package containing only features available under the Apache 2.0 license. To install this package, use:

```
echo "deb https://artifacts.elastic.co/packages/oss-7.x/apt stable
main" |
sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
```
Copy
All that's left to do is to update your repositories and install Kibana:

```
sudo apt-get update
sudo apt-get install kibana
```
Copy
Open up the Kibana configuration file at: */etc/kibana/kibana.yml*, and make sure you have the following configurations defined:

```
server.port: 5601
elasticsearch.url: "http://localhost:9200"
```
Copy
These specific configurations tell Kibana which Elasticsearch to connect to and which port to use.
Now, start Kibana with:

```
sudo service kibana start
```
Copy

Open up Kibana in your browser with: http://<yourServerIP>:5601. You will be presented with the Kibana home page.

Defining an index pattern

Your next step is to define a new index pattern, or in other words, tell Kibana what Elasticsearch index to analyze. To do that you will of course need to have data indexed. For the purpose of this tutorial, we've prepared some sample data containing Apache access logs that is refreshed daily. You can download the data here: https://logz.io/sample-data. Of course, you could send in your own logs .
Next, we will use Logstash to collect, parse and ship this data into Elasticsearch. If you haven't installed Logstash yet, or are not familiar with how to use it, check out this Logstash tutorial.
Create a new Logstash configuration file at: /etc/logstash/conf.d/apache-01.conf:

```
sudo vim /etc/logstash/conf.d/apache-01.conf
```

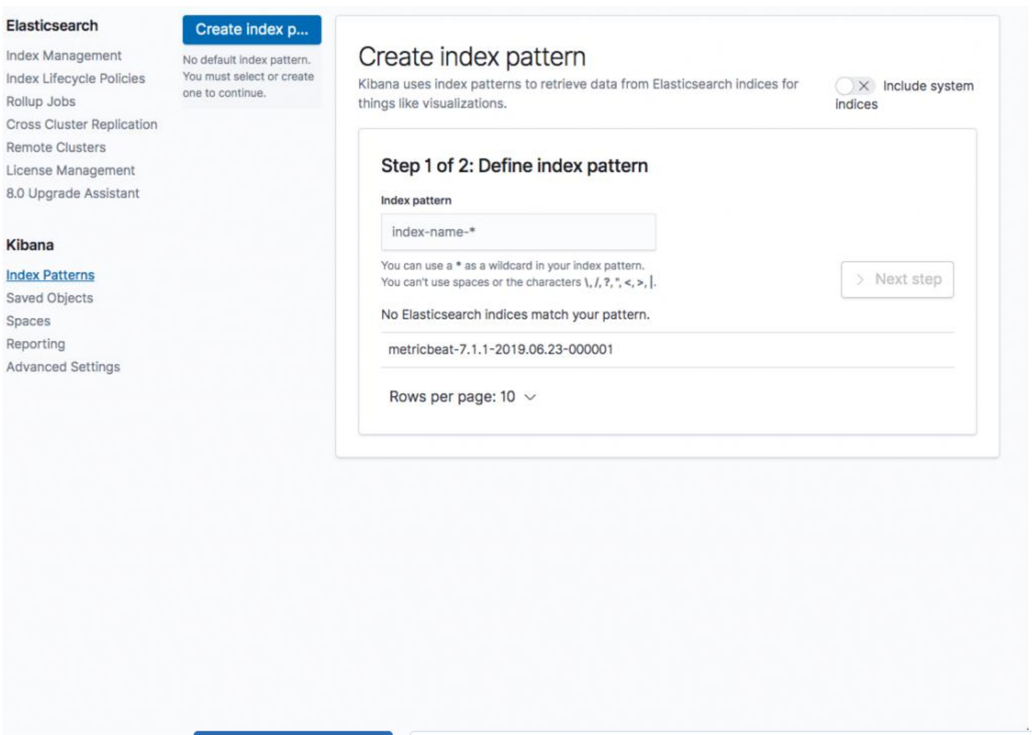
Copy
Enter the following Logstash configuration (change the path to the file you downloaded accordingly):

```
input {
  file {
    path => "/home/ubuntu/apache-daily-access.log"
    start_position => "beginning"
    sincedb_path => "/dev/null"
  }
}
filter {
    grok {
      match => { "message" => "%{COMBINEDAPACHELOG}" }
    }
    date {
    match => [ "timestamp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
  }
    geoip {
      source => "clientip"
    }
}
output {
  elasticsearch {
    hosts => ["localhost:9200"]
  }
}
```

Copy
Start Logstash with:

```
sudo service logstash start
```

Copy



Elasticsearch

Index Management
Index Lifecycle Policies
Rollup Jobs
Cross Cluster Replication
Remote Clusters
License Management
8.0 Upgrade Assistant

Kibana

Index Patterns
Saved Objects
Spaces
Reporting
Advanced Settings

Create index p...

No default index pattern. You must select or create one to continue.

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

Include system indices

Step 1 of 2: Define index pattern

Index pattern

index-name-*

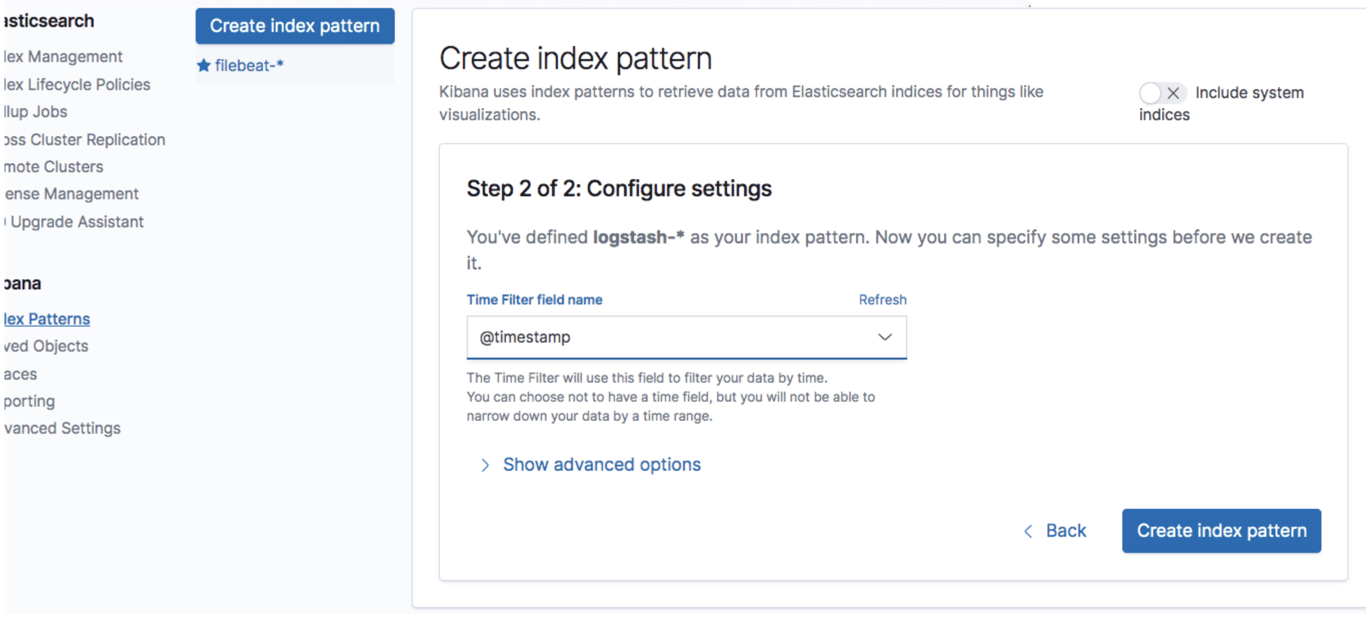
You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, |.

Next step

No Elasticsearch indices match your pattern.

metricbeat-7.1.1-2019.06.23-000001

Rows per page: 10



ısticsearch

lex Management
lex Lifecycle Policies
llup Jobs
oss Cluster Replication
mote Clusters
ense Management
Upgrade Assistant

oana

lex Patterns
ved Objects
aces
porting
vanced Settings

Create index pattern

★ filebeat-*

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

Include system indices

Step 2 of 2: Configure settings

You've defined **logstash-*** as your index pattern. Now you can specify some settings before we create it.

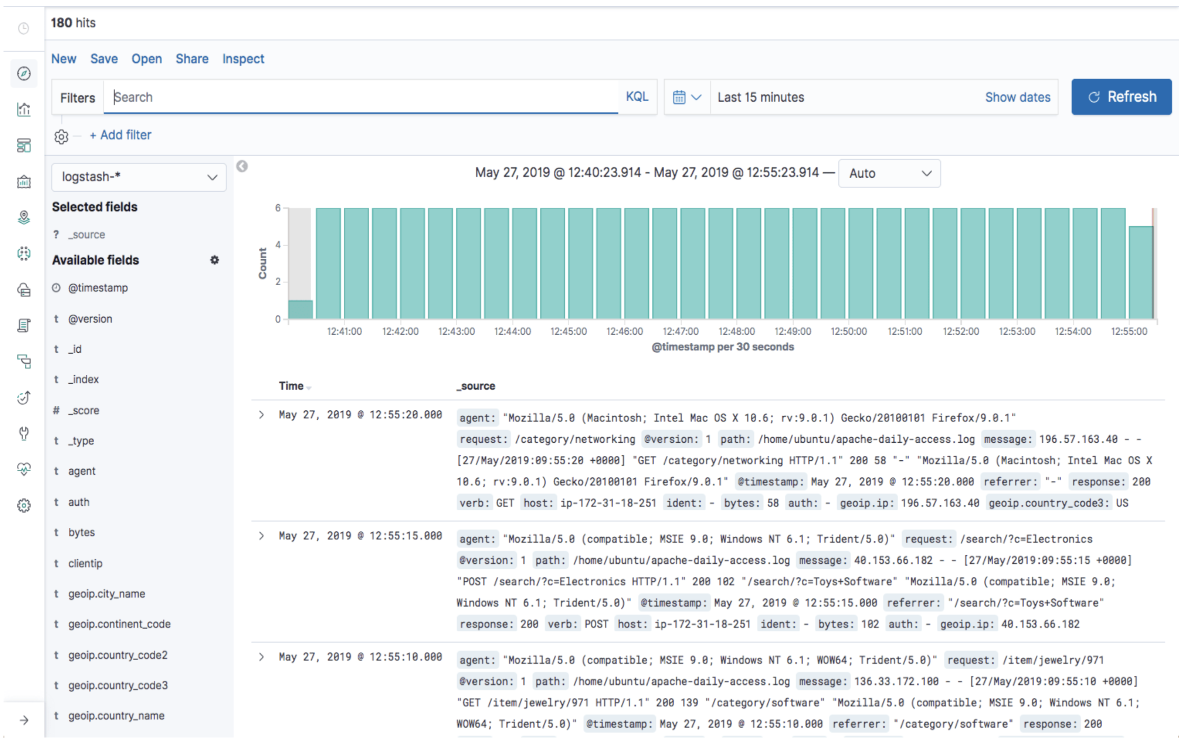
Time Filter field name — Refresh

@timestamp

The Time Filter will use this field to filter your data by time.
You can choose not to have a time field, but you will not be able to narrow down your data by a time range.

Show advanced options

Back — Create index pattern



```
> May 27, 2019 @ 12:55:20.000  agent: "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:9.0.1) Gecko/20100101 Firefox/9.0.1"
                               request: /category/networking @version: 1 path: /home/ubuntu/apache-daily-access.log message: 196.57.163.40 - -
                               [27/May/2019:09:55:20 +0000] "GET /category/networking HTTP/1.1" 200 58 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X
                               10.6; rv:9.0.1) Gecko/20100101 Firefox/9.0.1" @timestamp: May 27, 2019 @ 12:55:20.000 referrer: "-" response: 200
                               verb: GET host: ip-172-31-18-251 ident: - bytes: 58 auth: - geoip.ip: 196.57.163.40 geoip.country_code3: US

> May 27, 2019 @ 12:55:15.000  agent: "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)" request: /search/?c=Electronics
                               @version: 1 path: /home/ubuntu/apache-daily-access.log message: 40.153.66.182 - - [27/May/2019:09:55:15 +0000]
                               "POST /search/?c=Electronics HTTP/1.1" 200 102 "/search/?c=Toys+Software" "Mozilla/5.0 (compatible; MSIE 9.0;
                               Windows NT 6.1; Trident/5.0)" @timestamp: May 27, 2019 @ 12:55:15.000 referrer: "/search/?c=Toys+Software"
                               response: 200 verb: POST host: ip-172-31-18-251 ident: - bytes: 102 auth: - geoip.ip: 40.153.66.182

> May 27, 2019 @ 12:55:10.000  agent: "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)" request: /item/jewelry/971
                               @version: 1 path: /home/ubuntu/apache-daily-access.log message: 136.33.172.100 - - [27/May/2019:09:55:10 +0000]
                               "GET /item/jewelry/971 HTTP/1.1" 200 139 "/category/software" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1;
                               WOW64; Trident/5.0)" @timestamp: May 27, 2019 @ 12:55:10.000 referrer: "/category/software" response: 200
```

If all goes well, a new index will be created in Elasticsearch, the pattern of which can now be defined in Kibana.
In Kibana, go to **Management → Kibana Index Patterns**, and Kibana will automatically identify the new "logstash-*" index pattern.

Define it as "logstash-*", and in the next step select @timestamp as your Time Filter field.

Hit **Create index pattern**, and you are ready to analyze the data. Go to the Discover tab in Kibana to take a look at the data (look at today's data instead of the default last 15 mins).

Using Kibana in Logz.io

If you're using Logz.io, simply use this cURL command to upload the sample log data. Logz.io listeners will parse the data using automatic parsing so there's no need to configure Logstash (the token can be found on the Settings page in the Logz.io UI, and the type of the file is *apache_access*):

```
curl -T <Full path to file>
http://listener.logz.io:8021/file_upload/<Token>/apache_access
```
Copy

Kibana Searching

Kibana querying is an art unto itself, and there are various methods for performing searches on your data. This section will describe some of the most common search methods as well as some tips and best practices that should be memorized for optimized user experience.

KQL and Lucene

Up until version 6.2, the only way to query in Kibana was using Lucene syntax. Starting in version 6.2, another query language was introduced called Kuery, or as it’s been called now—KQL (Kibana Querying Language) to improve the searching experience.
Since version 7.0, KQL is the default language for querying in Kibana but you can revert to Lucene if you like. For the basic example below, there will be little difference in the search results.

Free-Text Search

Free text search works within all fields — including the _source field, which includes all the other fields. If no specific field is indicated in the search, the search will be done on all of the fields that are being analyzed.
In the search field at the top of the Discover page, run these searches and examine the result (set the time parameter on the top right of the dashboard to the past month to capture more data):

- category
- Category
- categ
- cat*
- categ?ry
- “category”
- category\/health
- “category/health”
- Chrome
- chorm*

Tips and gotchas

Not case sensitive

Text searches are not case sensitive. This means that *category* and *CaTeGory* will return the same results. When you put the text within double quotes (“”), you are looking for an exact match, which means that the exact string must match what is inside the double quotes. This is why [category\/health] and [“category/health”] will return different results

Wildcard symbols and search

Kibana wildcard searches – you can use the wildcard symbols [*] or [?] in searches. [*] means any number of characters, and [?] means only one character

Field-Level Searches

Another common search in Kibana is field-level queries, used for searching for data inside specific fields. To use this type of search that, you need to use the following format:

```
<fieldname>:search
```
Copy
As before, run the following searches to see what you get (some will purposely return no results):

- name:chrome
- name:Chrome
- name:Chr*
- response:200
- bytes:65
- bytes:[65 TO *]
- bytes:[65 TO 99]
- bytes:{65 TO 99}
- _exists_:name

Tips and gotchas

Field-type dependency

1. Field-level searches depend on the type of field. (Logz.io users – by default, *none* of the fields are analyzed, which means that searches in Logz.io *are* case-sensitive (unlike free-text search) and *cannot* use wildcard searches (again, unlike free-text search). The reason we save all of the fields as “not analyzed” is in order to save space in the index since the data is also duplicated in an analyzed field called *_source*)

Searching a range

1. You can search a range within a field. If you use [], this means that the results are inclusive. If you use {}, this means that the results are exclusive.

exists

1. Using the *_exists_* prefix for a field will search the documents to see if the field exists

Capitalization and ranges

1. When using a range, you need to follow a very strict format and use capital letters TO to specify the range

Logical Statements

You can use logical statements in searches in these ways:

- USA AND Firefox
- USA OR Firefox
- (USA AND Firefox) OR Windows
- -USA
- !USA
- +USA
- NOT USA

Tips and gotchas

1. You need to make sure that you use the proper format such as capital letters to define logical terms like AND or OR
2. You can use parentheses to define complex, logical statements
3. You can use -,! and NOT to define negative terms

Kibana special characters

All special characters need to be properly escaped. The following is a list of all available special characters:

+ – && || ! ( ) { } [ ] ^ ” ~ * ? : \

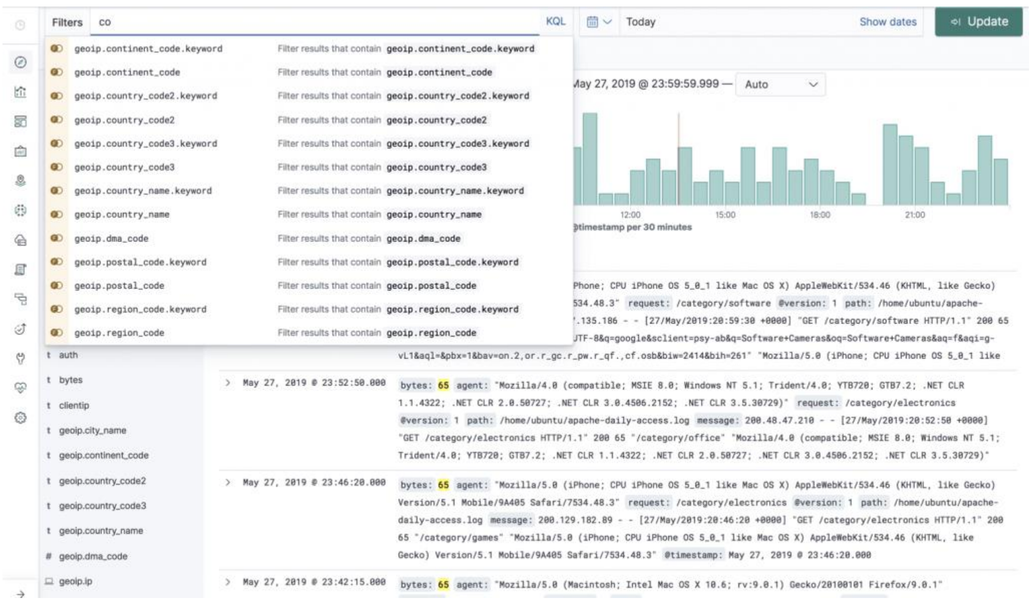
Proximity searches

Proximity searches are an advanced feature of Kibana that takes advantage of the Lucene query language.
[categovi~2] means a search for all the terms that are within two changes from [categovi]. This means that all category will be matched.

Tips and gotchas

Proximity searches use a lot of system resources and often trigger internal circuit breakers in Elasticsearch. If you try something such as [catefujt~10], it is likely not to return any results due to the amount of memory used to perform this specific search.

Kibana Autocomplete

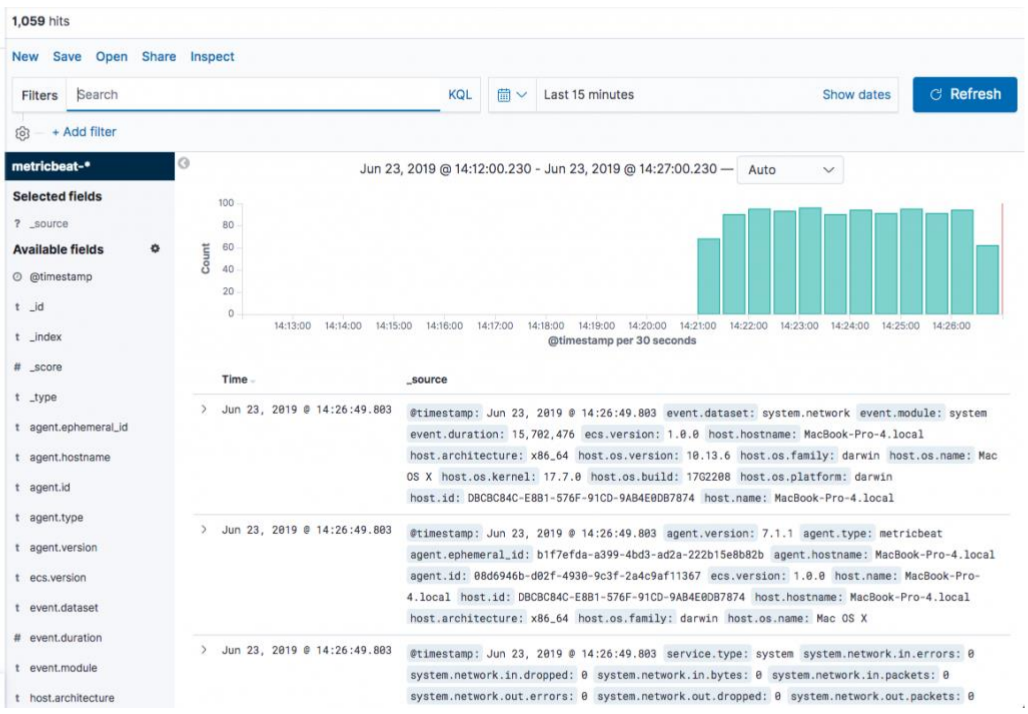


geoip.continent_code.keyword	Filter results that contain geoip.continent_code.keyword
geoip.continent_code	Filter results that contain geoip.continent_code
geoip.country_code2.keyword	Filter results that contain geoip.country_code2.keyword
geoip.country_code2	Filter results that contain geoip.country_code2
geoip.country_code3.keyword	Filter results that contain geoip.country_code3.keyword
geoip.country_code3	Filter results that contain geoip.country_code3
geoip.country_name.keyword	Filter results that contain geoip.country_name.keyword
geoip.country_name	Filter results that contain geoip.country_name
geoip.dma_code	Filter results that contain geoip.dma_code
geoip.postal_code.keyword	Filter results that contain geoip.postal_code.keyword
geoip.postal_code	Filter results that contain geoip.postal_code
geoip.region_code.keyword	Filter results that contain geoip.region_code.keyword
geoip.region_code	Filter results that contain geoip.region_code

```
Phone; CPU iPhone OS 5_0_1 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko)
534.48.3" request: /category/software @version: 1 path: /home/ubuntu/apache-
'.135.186 - - [27/May/2019:20:59:30 +0000] "GET /category/software HTTP/1.1" 200 65
JTF-8&q=google&sclient=psy-ab&q=Software+Cameras&oq=Software+Cameras&aq=f&aqi=g-
vL1&aql=&pbx=1&bav=on.2,or.r_gc.r_pw.r_qf.,cf.osb&biw=2414&bih=261" "Mozilla/5.0 (iPhone; CPU iPhone OS 5_0_1 like

> May 27, 2019 @ 23:52:50.000  bytes: 65 agent: "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; YTB720; GTB7.2; .NET CLR
                               1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)" request: /category/electronics
                               @version: 1 path: /home/ubuntu/apache-daily-access.log message: 200.48.47.210 - - [27/May/2019:20:52:50 +0000]
                               "GET /category/electronics HTTP/1.1" 200 65 "/category/office" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;
                               Trident/4.0; YTB720; GTB7.2; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)"

> May 27, 2019 @ 23:46:20.000  bytes: 65 agent: "Mozilla/5.0 (iPhone; CPU iPhone OS 5_0_1 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko)
                               Version/5.1 Mobile/9A405 Safari/7534.48.3" request: /category/electronics @version: 1 path: /home/ubuntu/apache-
                               daily-access.log message: 200.129.182.89 - - [27/May/2019:20:46:20 +0000] "GET /category/electronics HTTP/1.1" 200
                               65 "/category/games" "Mozilla/5.0 (iPhone; CPU iPhone OS 5_0_1 like Mac OS X) AppleWebKit/534.46 (KHTML, like
                               Gecko) Version/5.1 Mobile/9A405 Safari/7534.48.3" @timestamp: May 27, 2019 @ 23:46:20.000

> May 27, 2019 @ 23:42:15.000  bytes: 65 agent: "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:9.0.1) Gecko/20100101 Firefox/9.0.1"
```

To help improve the search experience in Kibana, the autocomplete feature suggests search syntax as you enter your query. As you type, relevant fields are displayed and you can complete the query with just a few clicks. This speeds up the whole process and makes Kibana querying a whole lot simpler.

Kibana Filtering



```
Jun 23, 2019 @ 14:26:49.803   @timestamp: Jun 23, 2019 @ 14:26:49.803 event.dataset: system.network event.module: system
                              event.duration: 15,702,476 ecs.version: 1.0.0 host.hostname: MacBook-Pro-4.local
                              host.architecture: x86_64 host.os.version: 10.13.6 host.os.family: darwin host.os.name: Mac
                              OS X host.os.kernel: 17.7.0 host.os.build: 17G2208 host.os.platform: darwin
                              host.id: DBCBC84C-E8B1-576F-91CD-9AB4E0DB7874 host.name: MacBook-Pro-4.local

Jun 23, 2019 @ 14:26:49.803   @timestamp: Jun 23, 2019 @ 14:26:49.803 agent.version: 7.1.1 agent.type: metricbeat
                              agent.ephemeral_id: b1f7efda-a399-4bd3-ad2a-222b15e8b82b agent.hostname: MacBook-Pro-4.local
                              agent.id: 08d6946b-d02f-4930-9c3f-2a4c9af11367 ecs.version: 1.0.0 host.name: MacBook-Pro-
                              4.local host.id: DBCBC84C-E8B1-576F-91CD-9AB4E0DB7874 host.hostname: MacBook-Pro-4.local
                              host.architecture: x86_64 host.os.family: darwin host.os.name: Mac OS X

Jun 23, 2019 @ 14:26:49.803   @timestamp: Jun 23, 2019 @ 14:26:49.803 service.type: system system.network.in.errors: 0
                              system.network.in.dropped: 0 system.network.in.bytes: 0 system.network.in.packets: 0
                              system.network.out.errors: 0 system.network.out.dropped: 0 system.network.out.packets: 0
```

To assist users in searches, Kibana includes a filtering dialog that allows easier filtering of the data displayed in the main view.
To use the dialog, simply click the **Add a filter +** button under the search box and begin experimenting with the conditionals. Filters can be pinned to the Discover page, named using custom labels, enabled/disabled and inverted.

Power users can also enter Elasticsearch queries using the Query DSL.

Thousands of engineers rely on Kibana's search and visualization capabilities to troubleshoot their environments.

For those who prefer Kibana, but also require more advanced log analysis capabilities – such as anomaly detection, alerting, ML to automatically surface critical exceptions, and correlation with metrics and traces – Logz.io Log Management may be the better option. Logz.io adds these capabilities on top of OpenSearch Dashboards to accelerate log analysis and troubleshooting.

Summary

This tutorial covered the basics of setting up and using Kibana and provided the steps for setting up a test environment only. Tracking Apache access logs in production, for example, is better done using Filebeat and the supplied Apache module. The sample data provided can, of course, be replaced with other types of data, as you see fit. Kibana also provides sets of sample data to play around with, including flight data and web logs.
If you have any suggestions on what else should be included in the first part of this Kibana tutorial, please let me know in the comments below. The next Kibana tutorial will cover visualizations and dashboards.

Enjoy!