

# LAB 3

## はじめに

本 Lab では、どのように複数のログファイルを Elastic Stack に投入するかを経験します。

## VM へのインストール

### 概要

Beats エージェントは軽量のデータシッパーとして設計されています。各 beat はそれぞれ特定のデータセットを扱います。本 Lab では、CPU やメモリー使用率を Elasticsearch に送信する Metricbeat と、NGINX や Apache といったサービスのログファイルを送信するだけでなく、システムの認証ログなども送信する Filebeat を使います。

### ソフトウェアダウンロード

Software	URL
Filebeat	<a href="https://www.elastic.co/downloads/beats/filebeat">https://www.elastic.co/downloads/beats/filebeat</a>

## Linux インストラクション

### Filebeat

- 1) ターミナルを開いて、filebeat をダウンロードします。

```
curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.4.2-linux-x86_64.tar.gz
```

- 2) ダウンロードしたファイルを展開します。

```
tar xzvf filebeat-7.4.2-linux-x86_64.tar.gz
```

- 3) filebeat に移動します。

```
cd filebeat-7.4.2-linux-x86_64
```

- 4) 利用可能なモジュールをリストします。

```
./filebeat modules list
```

どのモジュールが **enabled** で、どのモジュールが **disabled** かを確認します。デフォルトでは全てのモジュールが利用不可となっています。

```
# ./filebeat modules list
Enabled:
  4) List models that are available

Disabled:
apache2
auditd
elasticsearch
haproxy
icinga
iis
kafka
kibana
logstash
mongodb
mysql
nginx
osquery
postgresql
redis
suricata
system
traefik
```

`./filebeat modules list`

You should see which modules are *enabled* and which modules are *disabled*. If there are no modules enabled.

5) NGINX logs を投入するために NGINX module を enable にします。

```
./filebeat modules enable nginx
```

```
# ./filebeat modules list
Enabled:
nginx

Disabled:
apache2
auditd
elasticsearch
haproxy
icinga
iis
kafka
kibana
logstash
mongodb
mysql
nginx
osquery
postgresql
redis
suricata
system
traefik
```

You should see which modules are *enabled* and which modules are *disabled*. If there are no modules enabled.

5) Now let us enable the NGINX module so we can ingest NG

6) Elasticsearch が NGINX logs を受け入れるようにする前に、Filebeat に Elasticsearch が何処にあるかと認証情報を教えてあげる必要があります。Filebeat の構成ファイルである filebeat.yml を編集します。

お好きなテキストエディタで `filebeat.yml` を開き、**cloud.id** と **cloud.auth** を Lab 0 で取得した値に変更します。



YAML files don't like hard tabs. Do not use them if you are editing a .yml file because they will cause errors. To learn more about .yml files see this link: <https://en.wikipedia.org/wiki/YAML>

例 : (以下は例ですので、**実際にはご自身のものをお使いください**)

```
#cloud.id:
```

を以下のように変更

```
cloud.id: "以下のクラウドコンソールからコピー"
```

cloud.id は、Lab0 のクラウドコンソールの自身の Deployment からコピーします。

The screenshot shows the Elastic Cloud console interface. On the left is a navigation menu with sections: Deployments, Custom plugins, Account, and Help. Under Deployments, 'Workshop' is selected. The main content area shows details for the 'Workshop' deployment (ID: 7bb1a3). It includes a 'Deployment name' field with 'Workshop' and a 'Rename deployment' button. The 'Deployment version' is 'v6.5.2'. The 'Deployment status' is 'Success' (indicated by a green checkmark). Under 'Endpoints', 'Elasticsearch' and 'Kibana' are listed. A 'Cloud ID' field is highlighted with a red box, containing the value: 'Workshop: dXMtZWFzdC0xLmF3cy5mb3VuZC5pbyQ3YmIxYTM5OWYwODk0OTEzYWU3M2ExNWVjNzI2MjdiZCQyOGJkZTNhMzY4ZjM0ODViODJhMDM1M2QxMjlmNWU0Yw=='. A help icon is visible next to the Cloud ID label.

```
#cloud.auth:
```

を以下のように変更。"elastic:"はユーザー名と区切り文字です。パスワードを":"より後ろに入力します。

```
cloud.auth: "elastic:Lab0 の Step14 でコピーしたパスワード"
```

- 通常ならば Filebeat はインストールされたマシンの所定のフォルダから NGINX ログファイルをスキャンします。今回は、NGINX をインストールしていないため、実際の NGINX ログファイルを Laptop のファイルシステムにコピーして、そのフォルダを NGINX モジュールに教えてあげることにします。

以下の URL から NGINX logs をダウンロードし、展開します：

```
wget https://lgworkshop-bucket.s3.amazonaws.com/nginx.log.zip  
  
unzip nginx.log.zip
```

- 以下のディレクトリに移動します。

```
cd modules.d
```

- NGINX module がダウンロードしたログファイルの場所を参照するようにします。nginx.yml ファイルを編集し、ダウンロードしたログファイルのためのエントリを追加します。以下の設定を nginx.yml に追加します。

**Note: NGINX log ファイルを展開した場所を指すように修正します。**  
(例："/home/ubuntu/nginx.log")

```
- module: nginx  
  # Access logs  
  access:  
    enabled: true  
    # Set custom paths for the log files. If left empty,  
    # Filebeat will choose the paths depending on your OS.  
    #var.paths:  
    var.paths: ["/Users/shh/Development/logs/nginx/nginx.log"]  
  
  # Error logs  
  error:  
    enabled: true  
    # Set custom paths for the log files. If left empty,  
    # Filebeat will choose the paths depending on your OS.  
    #var.paths:
```

- Elasticsearch が NGINX logs を受け取り、可視化し、異常検知するための機械学習ジョブを作成する準備が整いました。次のコマンドを実行します。しばらく時間がかかるので放置しておきます。

```

cd ..
./filebeat -e setup nginx
./filebeat -e

```

```

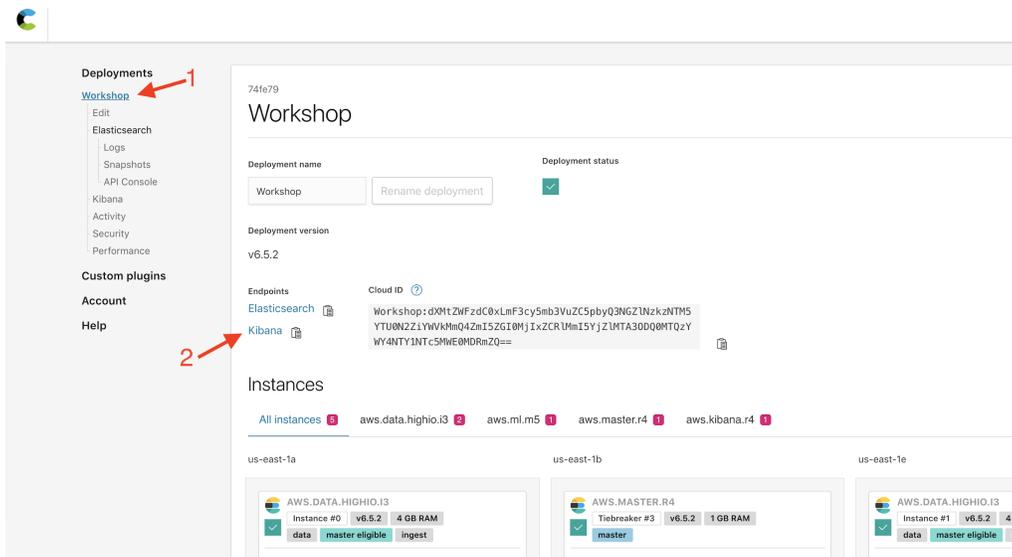
# ./filebeat -e setup nginx
2018-12-11T01:32:03.831-0500 INFO instance/beat.go:592 Home path: [/Users/shh/Development/filebeat-6.5.2-darwin-x86_64] Config path: [/Users/shh/Development/filebeat-6.5.2-darwin-x86_64] Data path: [/Users/shh/Development/filebeat-6.5.2-darwin-x86_64/data] Logs path: [/Users/shh/Development/filebeat-6.5.2-darwin-x86_64/logs]
2018-12-11T01:32:03.831-0500 INFO instance/beat.go:599 Beat UUID: 3c16d505-9652-42d7-b7c2-547ad985cb1d
2018-12-11T01:32:03.837-0500 INFO [beat] instance/beat.go:825 Beat info {"system_info": {"beat": {"path": {"config": "/Users/shh/Development/filebeat-6.5.2-darwin-x86_64", "data": "/Users/shh/Development/filebeat-6.5.2-darwin-x86_64/data", "home": "/Users/shh/Development/filebeat-6.5.2-darwin-x86_64", "logs": "/Users/shh/Development/filebeat-6.5.2-darwin-x86_64/logs", "type": "filebeat", "uuid": "3c16d505-9652-42d7-b7c2-547ad985cb1d"}}}}
2018-12-11T01:32:03.838-0500 INFO [beat] instance/beat.go:834 Build info {"system_info": {"build": {"commit": "b48d073b84e874a182c122d8ef2bad867f714a11", "libbeat": "6.5.2", "time": "2018-11-29T23:03:04.000Z", "version": "6.5.2"}}}
2018-12-11T01:32:03.837-0500 INFO [beat] instance/beat.go:837 Go runtime info {"system_info": {"go": {"os": "darwin", "arch": "amd64", "max_procs": 8, "version": "go1.10.3"}}}
2018-12-11T01:32:03.839-0500 INFO [beat] instance/beat.go:841 Host info {"system_info": {"host": {"architecture": "x86_64", "boot_time": "2018-12-07T13:04:35.829985-05:00", "name": "Shawns-MacBook-Pro-2.local", "ip": ["127.0.0.1/8", "::1/128", "fe80::1/64", "192.168.1.13/24", "fe80::f85a:d0ff:feaa:5047/64", "fe80::56c0:da23:8d5a:7418/64", "fe80::5258:6fd:32c3:ca2d/64", "fe80::aede:48ff:fe00:1122/64"], "kernel_version": "18.0.0", "mac": [{"8c:85:90:ad:2d:5e", "fa:5a:d0:aa:50:47", "ac:de:48:00:11:22"}], "os": {"family": "darwin", "platform": "darwin", "name": "Mac OS X", "version": "10.14", "major": 10, "minor": 14, "patch": 0, "build": "18A391"}, "timezone": "EST", "timezone_offset_sec": -18000}}}
2018-12-11T01:32:03.840-0500 INFO [beat] instance/beat.go:870 Process info {"system_info": {"process": {"cwd": "/Users/shh/Development/filebeat-6.5.2-darwin-x86_64", "exe": "/Users/shh/Development/filebeat-6.5.2-darwin-x86_64/bin/filebeat", "name": "filebeat", "pid": 5719, "ppid": 2954, "start_time": "2018-12-11T01:32:03.804-0500"}}}
2018-12-11T01:32:03.840-0500 INFO instance/beat.go:278 Setup Beat: filebeat; Version: 6.5.2
2018-12-11T01:32:06.845-0500 INFO add_cloud_metadata/add_cloud_metadata.go:319 add_cloud_metadata: hosting provider type not detected.
2018-12-11T01:32:06.851-0500 INFO elasticsearch/client.go:163 Elasticsearch url: https://dfbbe109f62b413188ae4f3e886355ef.us-east-1.aws.found.io:443
2018-12-11T01:32:06.852-0500 INFO [publisher] pipeline/module.go:110 Beat name: Shawns-MacBook-Pro-2.local
2018-12-11T01:32:06.853-0500 INFO elasticsearch/client.go:163 Elasticsearch url: https://dfbbe109f62b413188ae4f3e886355ef.us-east-1.aws.found.io:443
2018-12-11T01:32:07.250-0500 INFO elasticsearch/client.go:712 Connected to Elasticsearch version 6.5.2
2018-12-11T01:32:07.289-0500 INFO template/load.go:129 Template already exists and will not be overwritten.
Loaded index template
Loading dashboards (Kibana must be running and reachable)
2018-12-11T01:32:07.290-0500 INFO elasticsearch/client.go:163 Elasticsearch url: https://dfbbe109f62b413188ae4f3e886355ef.us-east-1.aws.found.io:443
2018-12-11T01:32:07.536-0500 INFO elasticsearch/client.go:712 Connected to Elasticsearch version 6.5.2
2018-12-11T01:32:07.536-0500 INFO kibana/client.go:118 Kibana url: https://ca131840ad5749fca8dadae599a42669.us-east-1.aws.found.io:443
2018-12-11T01:32:49.235-0500 INFO instance/beat.go:717 Kibana dashboards successfully loaded.
Loaded dashboards
2018-12-11T01:32:49.235-0500 INFO elasticsearch/client.go:163 Elasticsearch url: https://dfbbe109f62b413188ae4f3e886355ef.us-east-1.aws.found.io:443
2018-12-11T01:32:49.535-0500 INFO elasticsearch/client.go:712 Connected to Elasticsearch version 6.5.2
2018-12-11T01:32:49.535-0500 INFO kibana/client.go:118 Kibana url: https://ca131840ad5749fca8dadae599a42669.us-east-1.aws.found.io:443
Loaded machine learning job configurations

```

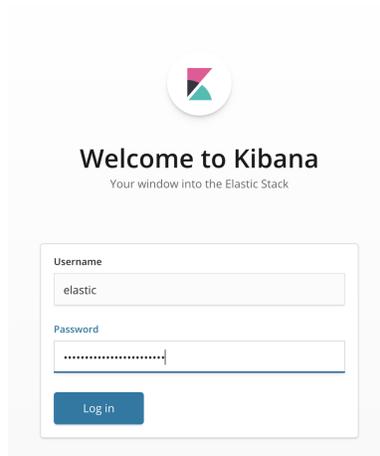
# Kibana でデータを確認

Kibana で Index を確認してみましょう。

- 1) クラウドコンソールにログインして、Kibana link をクリックします。

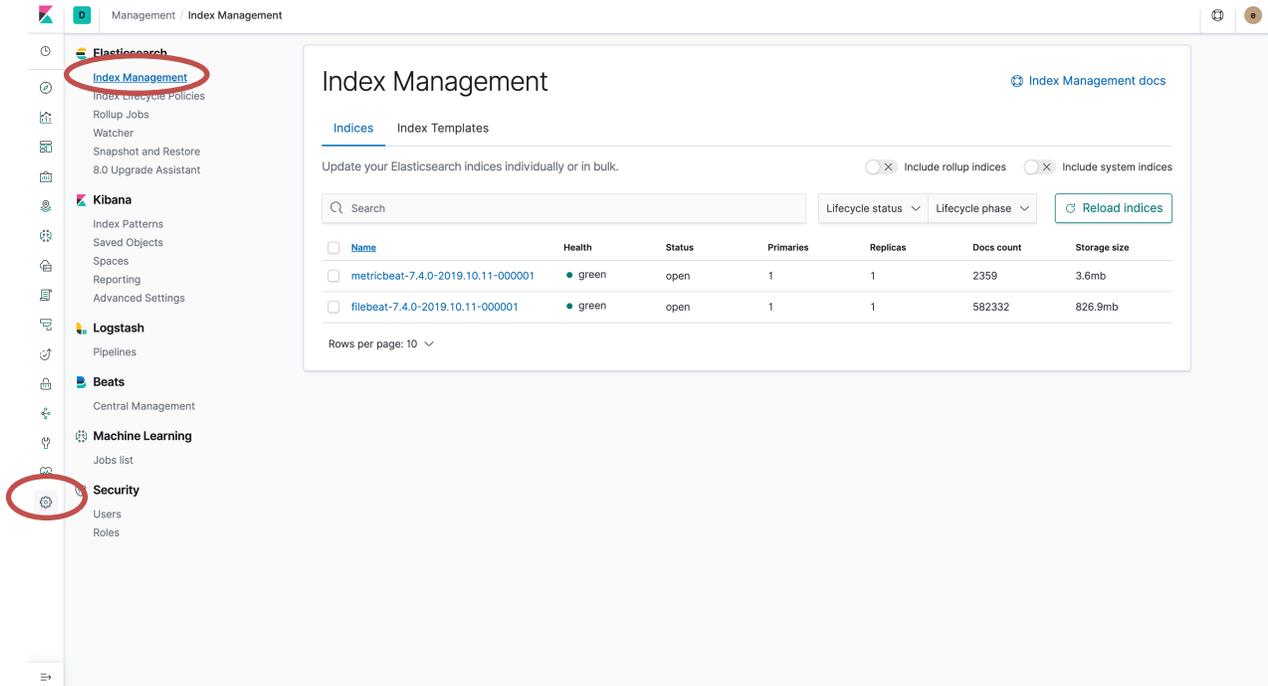


2) Lab 0 で取得したクレデンシャルで Kibana にログインします。



The image shows the Kibana login interface. At the top center is the Kibana logo, a circle containing a stylized 'K' with red, green, and blue segments. Below the logo, the text reads "Welcome to Kibana" in a bold font, followed by the tagline "Your window into the Elastic Stack" in a smaller font. The login form is enclosed in a light gray border and contains two input fields: "Username" with the value "elastic" and "Password" with a masked password represented by dots. A blue "Log in" button is positioned below the password field.

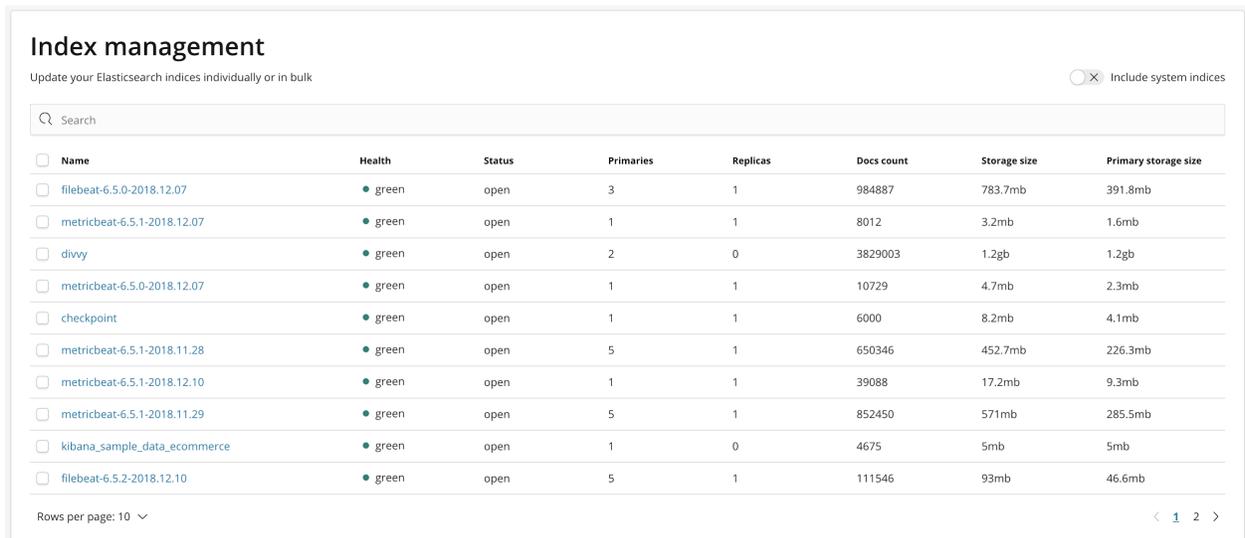
### 3) Management Link をクリックします。



The screenshot shows the Elastic Management console interface. In the left sidebar, the 'Index Management' link is circled in red. The main content area displays the 'Index Management' page, which includes a search bar, a table of indices, and a 'Reload indices' button. The table lists two indices: 'metricbeat-7.4.0-2019.10.11-000001' and 'filebeat-7.4.0-2019.10.11-000001'. Both indices are in a 'green' health state and 'open' status.

Name	Health	Status	Primaries	Replicas	Docs count	Storage size
<input type="checkbox"/> metricbeat-7.4.0-2019.10.11-000001	● green	open	1	1	2359	3.6mb
<input type="checkbox"/> filebeat-7.4.0-2019.10.11-000001	● green	open	1	1	582332	826.9mb

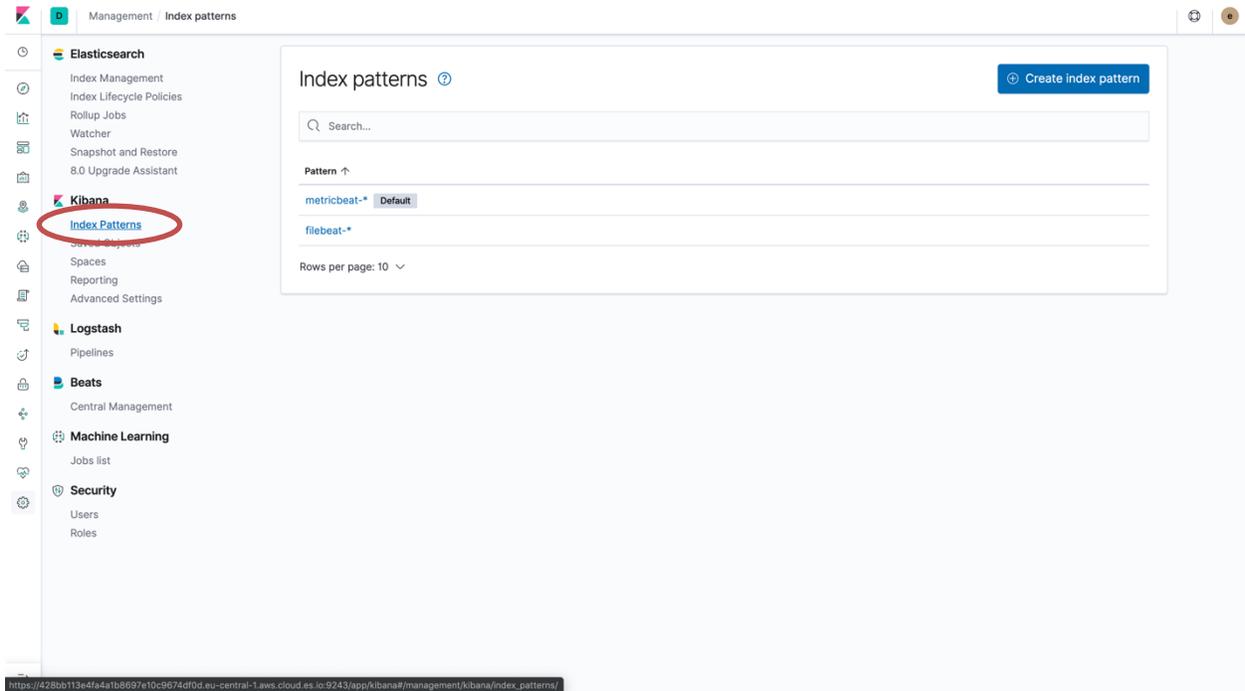
### 4) filebeat-<version>-YYYY.MM.DD-000001 という Index を探してみてください。version は製品のバージョン、YYYY, MM, DD は年月日を表します。Docs Count, Storage Size も確認してください。



The screenshot shows the Elastic Management console interface with a search filter applied. The search results show a table of indices, including 'filebeat-6.5.0-2018.12.07'. The table lists various indices with their health, status, and storage size.

Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Primary storage size
<input type="checkbox"/> filebeat-6.5.0-2018.12.07	● green	open	3	1	984887	783.7mb	391.8mb
<input type="checkbox"/> metricbeat-6.5.1-2018.12.07	● green	open	1	1	8012	3.2mb	1.6mb
<input type="checkbox"/> divvy	● green	open	2	0	3829003	1.2gb	1.2gb
<input type="checkbox"/> metricbeat-6.5.0-2018.12.07	● green	open	1	1	10729	4.7mb	2.3mb
<input type="checkbox"/> checkpoint	● green	open	1	1	6000	8.2mb	4.1mb
<input type="checkbox"/> metricbeat-6.5.1-2018.11.28	● green	open	5	1	650346	452.7mb	226.3mb
<input type="checkbox"/> metricbeat-6.5.1-2018.12.10	● green	open	1	1	39088	17.2mb	9.3mb
<input type="checkbox"/> metricbeat-6.5.1-2018.11.29	● green	open	5	1	852450	571mb	285.5mb
<input type="checkbox"/> kibana_sample_data_ecommerce	● green	open	1	0	4675	5mb	5mb
<input type="checkbox"/> filebeat-6.5.2-2018.12.10	● green	open	5	1	111546	93mb	46.6mb

- 5) 次に Kibana > Index Patterns をクリックしてください。Index patterns は Kibana に Elasticsearch のどの Index を探索したいのかを教えます。Index pattern は単一の Index の名前でも、wildcard(\*)を含む複数の Index でもマッチングさせることができます。



6) filebeat の Index patterns を確認してください。どの field が searchable か、aggregatable かを確認してください。

The screenshot shows the Kibana interface for the 'metricbeat-\*' index pattern. The left sidebar contains navigation options for Elasticsearch, Kibana, Logstash, Beats, Machine Learning, and Security. The main content area displays the index pattern 'metricbeat-\*' and a table of fields.

Time Filter field name: @timestamp | Default

This page lists every field in the **metricbeat-\*** index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the [Elasticsearch Mapping API](#).

Fields (2637) | Scripted fields (0) | Source filters (0)

Q Filter | All field types

Name	Type	Format	Searchable	Aggregatable	Excluded
@timestamp	date		●	●	✎
_id	string		●	●	✎
_index	string		●	●	✎
_score	number				✎
_source	_source				✎
_type	string		●	●	✎
aerospike.namespace.client.delete.error	number		●	●	✎
aerospike.namespace.client.delete.not_found	number		●	●	✎
aerospike.namespace.client.delete.success	number		●	●	✎
aerospike.namespace.client.delete.timeout	number		●	●	✎

Rows per page: 10 | < 1 2 3 4 5 ... 264 >



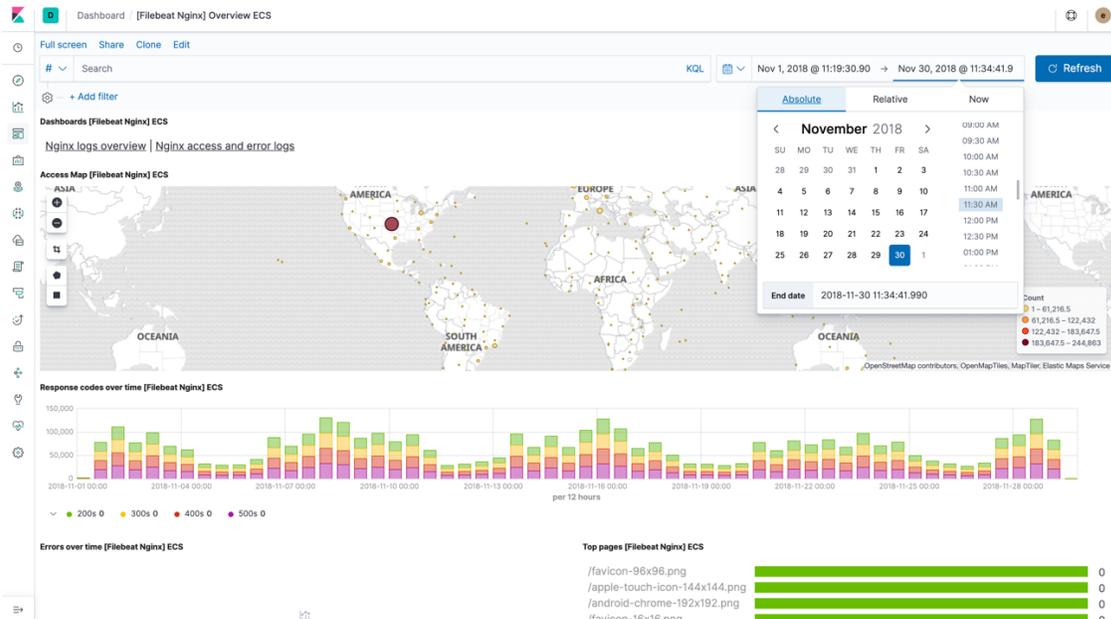
Timestamp	event.dataset	Message
Nov 24, 2018 @ 21:40:02.000	nginx.access	[nginx][access] 24.194.207.201 - "GET /fr/products HTTP/1.1" 200 75987
Nov 24, 2018 @ 21:49:51.000	nginx.access	[nginx][access] 124.182.150.239 - "GET /fr/products HTTP/1.1" 200 75987
Nov 24, 2018 @ 22:02:14.000	nginx.access	[nginx][access] 78.135.164.135 - "GET /fr/products HTTP/1.1" 200 75987
Nov 24, 2018 @ 22:54:12.000	nginx.access	[nginx][access] 3.246.17.110 - "GET /fr/products HTTP/1.1" 200 75987
Nov 24, 2018 @ 23:05:40.000	nginx.access	[nginx][access] 22.180.154.149 - "GET /fr/products HTTP/1.1" 200 75988
Nov 24, 2018 @ 23:43:24.000	nginx.access	[nginx][access] 176.61.87.81 - "GET /fr/products HTTP/1.1" 200 75987
Nov 24, 2018 @ 23:48:53.000	nginx.access	[nginx][access] 3.246.16.60 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 00:02:24.000	nginx.access	[nginx][access] 191.173.173.39 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 00:35:30.000	nginx.access	[nginx][access] 181.145.76.253 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 00:36:21.000	nginx.access	[nginx][access] 4.235.228.165 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 00:49:23.000	nginx.access	[nginx][access] 4.235.228.165 - "GET /fr/products HTTP/1.1" 200 75988
Nov 25, 2018 @ 00:50:26.000	nginx.access	[nginx][access] 15.83.226.80 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 00:56:45.000	nginx.access	[nginx][access] 177.145.23.27 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 01:03:18.000	nginx.access	[nginx][access] 0.180.2.178 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 01:12:28.000	nginx.access	[nginx][access] 7.126.6.178 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 01:12:28.000	nginx.access	[nginx][access] d1e6:e347:fe42:1cff:d3:fd00:6012:37f9 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 01:13:26.000	nginx.access	[nginx][access] 0.191.107.137 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 01:41:50.000	nginx.access	[nginx][access] 24.0.110.110 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 01:41:56.000	nginx.access	[nginx][access] 217.58.132.205 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 01:56:42.000	nginx.access	[nginx][access] 78.167.18.155 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 02:01:20.000	nginx.access	[nginx][access] 185.69.206.208 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 02:10:25.000	nginx.access	[nginx][access] 177.144.90.212 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 06:12:18.000	nginx.access	[nginx][access] 22.193.142.97 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 06:47:40.000	nginx.access	[nginx][access] 185.254.110.121 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 07:00:29.000	nginx.access	[nginx][access] 165.60.116.206 - "GET /fr/products HTTP/1.1" 200 75988
Nov 25, 2018 @ 12:15:32.000	nginx.access	[nginx][access] 169.252.91.210 - "GET /fr/products HTTP/1.1" 200 75987
Nov 25, 2018 @ 13:16:27.000	nginx.access	[nginx][access] 170.68.158.17 - "GET /fr/products HTTP/1.1" 200 75988

この機能は、検索エンジン(Elasticsearch)によるものです。検索の機能は、ログインでも非常に有効です。

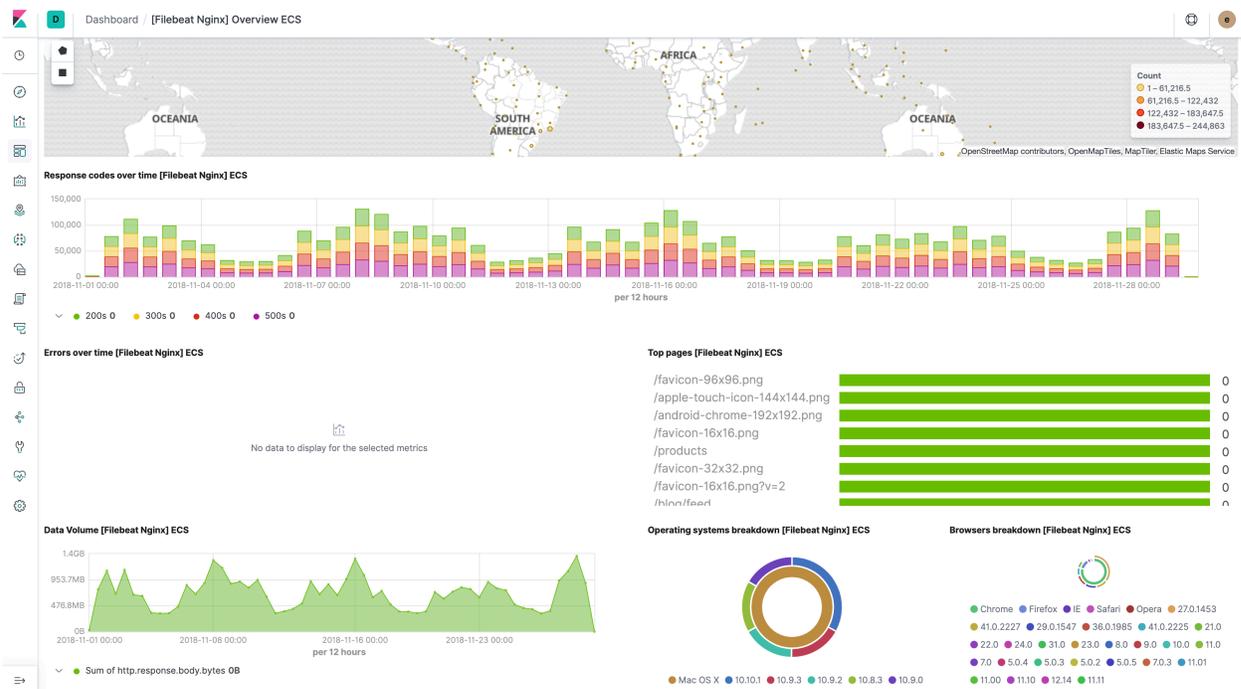
4. それでは、Filebeat のアウトオブボックスの Dashboards を見てみましょう。Menu の Dashboards をクリックします。全ての Dashboards のリストが表示されます。“Nginx”を検索してみます。“[Filebeat Nginx] Overview ECS” をクリックします。

Title	Description	Actions
<input type="checkbox"/> [Metricbeat Nginx] Overview ECS	Overview dashboard for the Nginx module in Metricbeat	
<input type="checkbox"/> [Filebeat Nginx] Overview ECS	Dashboard for the Filebeat Nginx module	
<input type="checkbox"/> [Filebeat Nginx] Access and error logs ECS	Dashboard for the Filebeat Nginx module	

Dashboard が開いたら、Time Picker で “Absolute” を選択し、between Nov 1<sup>st</sup> 2018 and Nov 28<sup>th</sup> 2018.としてください。

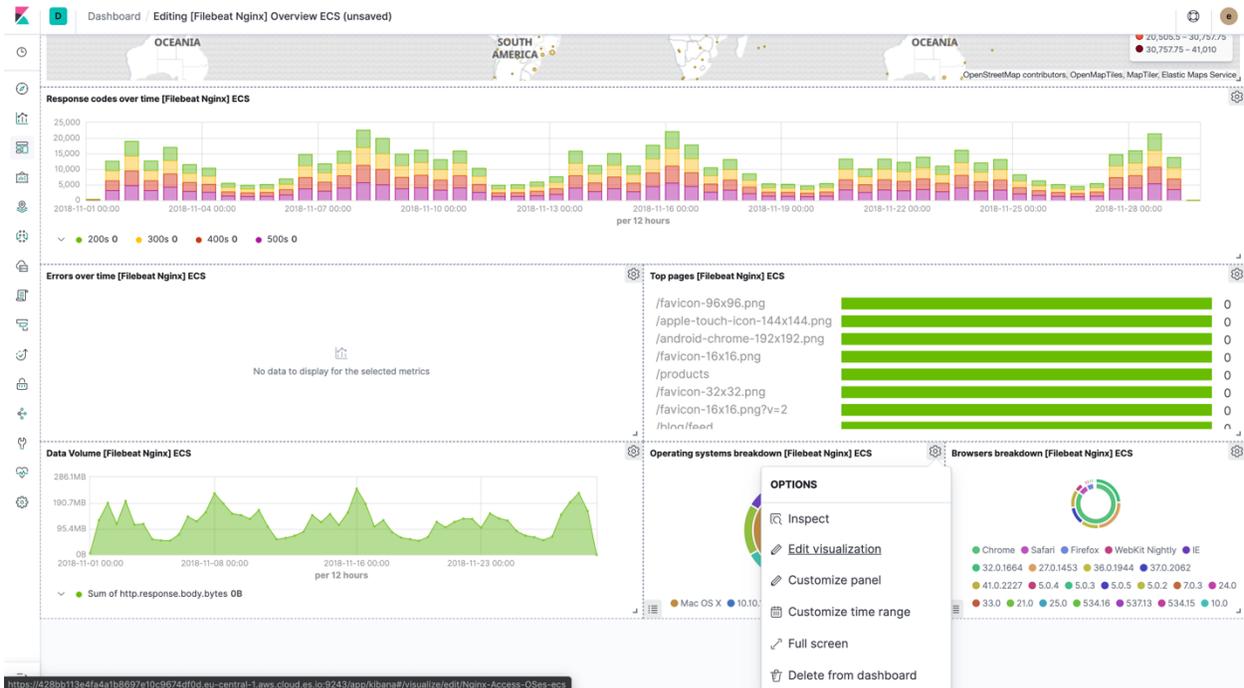


以下のような Dashboard が表示されるはずですが。

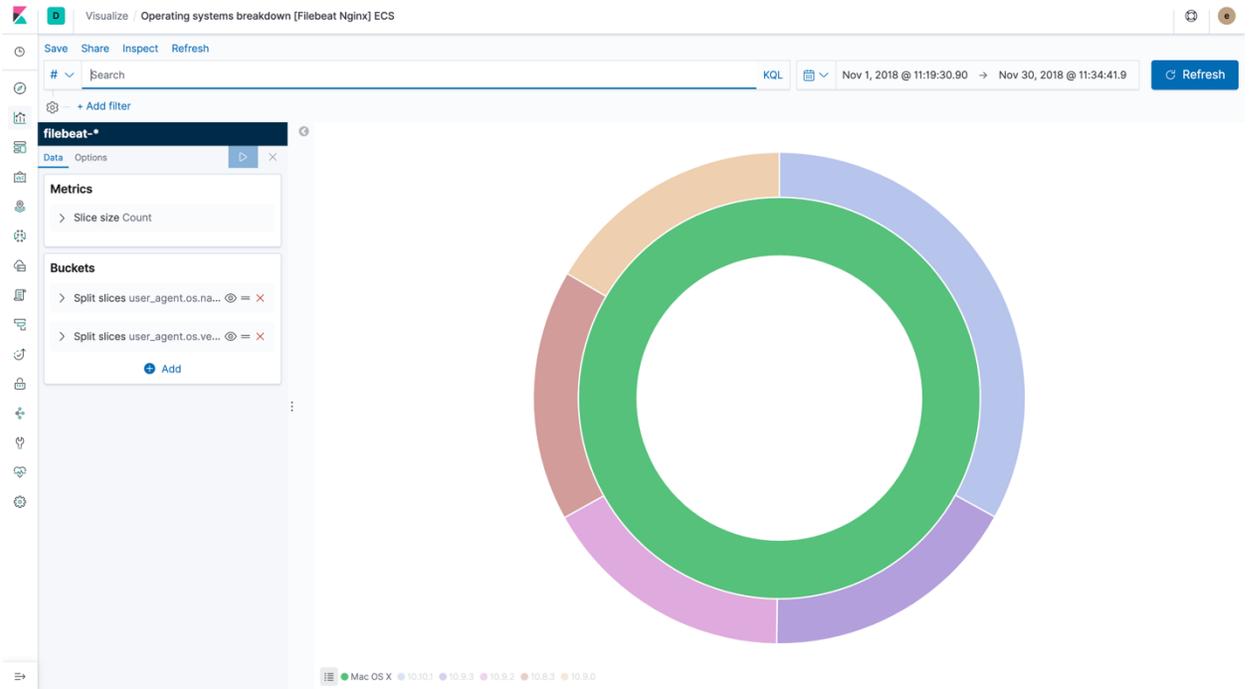


November 28<sup>th</sup> までのデータが全て見えない場合、まだデータがローディング中かも知れません。Auto-Refresh (next to date picker) を On にして dashboard がリアルタイムに更新されるのを見てみましょう。

5. これらの dashboards は Kibana でどのように visualization が作られるのかを見るいい例となります。“Edit” (next to Auto-Refresh option) をクリックし、visualization がどのようになっているか見てみましょう。



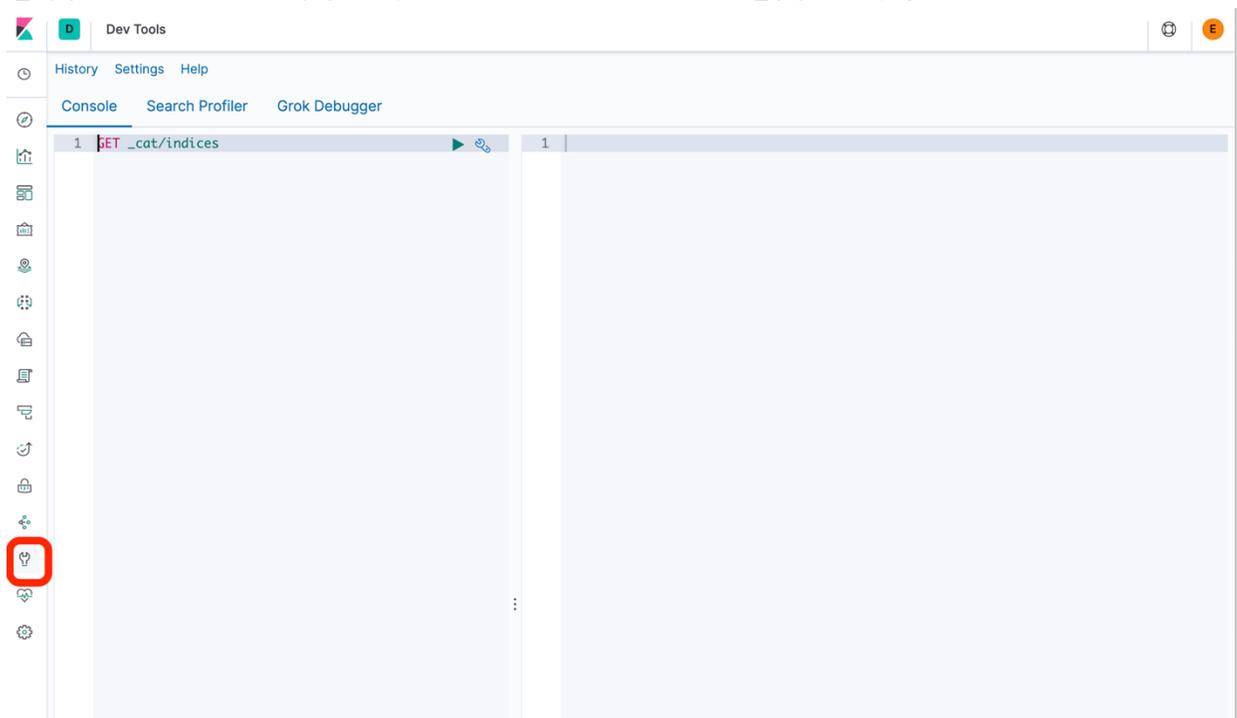
Pie chart visualization の例 :



# filebeat のモジュールの仕組みを確認

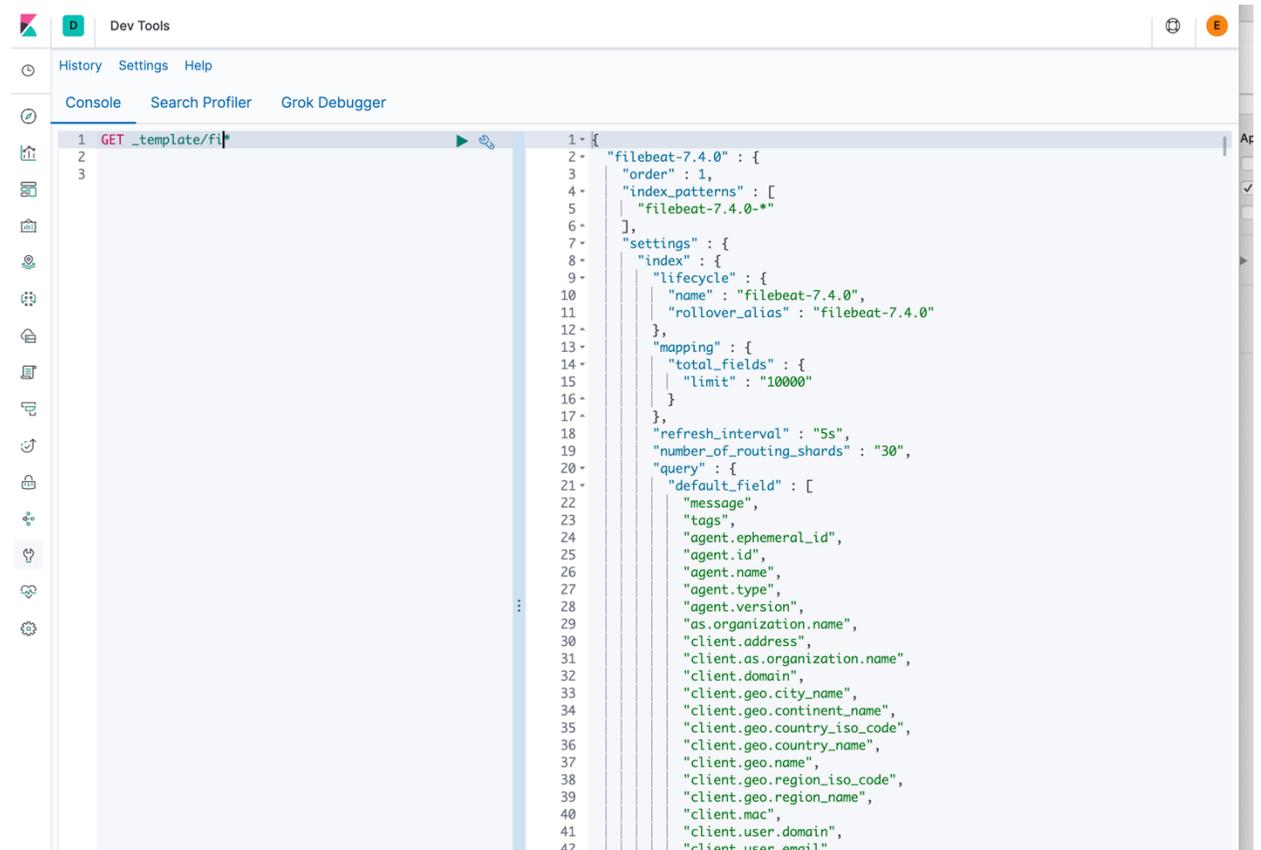
Filebeat のモジュールがどのような仕組みでログの文字列を構造化かしているのかを確認しましょう。

1. Index Template を確認しましょう。Dev Tools の Console を利用して、Elasticsearch にリクエストを簡単に送ることができます。Console を利用して、Index Template を確認してみましょう。まずは Dev Tools の Console を開きます。



2. Index Template の確認のために、"GET \_template/fi\*"と入力して、右側の緑の三角（再生）ボタンをクリックします。すると、右側に Elasticsearch からのレスポンス

スガ帰ってきます。

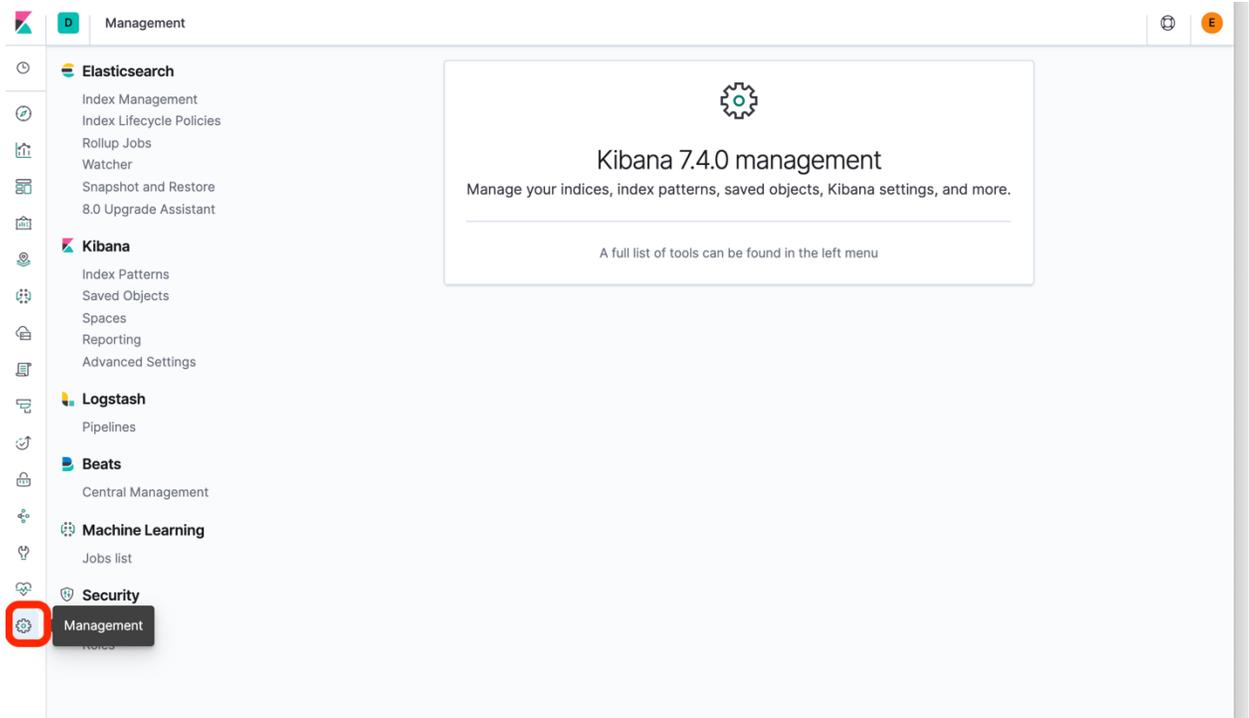


```
1 GET _template/fi
2
3

1 {
2   "filebeat-7.4.0" : {
3     "order" : 1,
4     "index_patterns" : [
5       "filebeat-7.4.0-*"
6     ],
7     "settings" : {
8       "index" : {
9         "lifecycle" : {
10          "name" : "filebeat-7.4.0",
11          "rollover_alias" : "filebeat-7.4.0"
12        },
13        "mapping" : {
14          "total_fields" : {
15            "limit" : "10000"
16          }
17        },
18        "refresh_interval" : "5s",
19        "number_of_routing_shards" : "30",
20        "query" : {
21          "default_field" : [
22            "message",
23            "tags",
24            "agent.ephemeral_id",
25            "agent.id",
26            "agent.name",
27            "agent.type",
28            "agent.version",
29            "as.organization.name",
30            "client.address",
31            "client.as.organization.name",
32            "client.domain",
33            "client.geo.city_name",
34            "client.geo.continent_name",
35            "client.geo.country_iso_code",
36            "client.geo.country_name",
37            "client.geo.name",
38            "client.geo.region_iso_code",
39            "client.geo.region_name",
40            "client.mac",
41            "client.user.domain",
42            "client.user.email".
```

- 次に Ingest Pipeline を確認しましょう。先ほど NGINX のログを Filebeat で取り込む設定をしました。Filebeat が Elasticsearch に送信した 1 行のログ文字列を、Elasticsearch がデータを取り込むタイミングでどのような処理を行っているかが、Ingest Pipeline を確認することでわかります。Dev Tools の Console で” GET \_ingest/pipeline/filebeat-\*nginx-access-\*”と入力して、実行しましょう。





- Kibana や Elasticsearch などの管理メニューにアクセスするための画面です。
- 次に、Kibana のメニューから Saved Objects クリックします。検索バーに“filebeat”を入力して検索してみましょう。

Management Saved objects

### Saved Objects

Export 351 objects Import Refresh

From here you can delete saved objects, such as saved searches. You can also edit the raw data of saved objects. Typically objects are only modified via their associated application, which is probably what you should use instead of this screen.

filebeat Type Delete Export

<input type="checkbox"/>	Type	Title	Actions
<input type="checkbox"/>		[Filebeat CoreDNS] Overview	...
<input type="checkbox"/>		[Filebeat Cisco] ASA Firewall	...
<input type="checkbox"/>		[Filebeat PostgreSQL] Overview ECS	...
<input type="checkbox"/>		[Filebeat IBM MQ] Overview of error log overview	...
<input type="checkbox"/>		[Filebeat Iptables] Overview ECS	...
<input type="checkbox"/>		[Filebeat Envoyproxy] Overview	...
<input type="checkbox"/>		[Filebeat MongoDB] Overview ECS	...
<input type="checkbox"/>		[Filebeat Kafka] Overview ECS	...
<input type="checkbox"/>		[Filebeat Iptables] Ubiquiti Firewall Overview ECS	...
<input type="checkbox"/>		[Filebeat Auditd] Audit Events ECS	...
<input type="checkbox"/>		[Filebeat Suricata] Events Overview ECS	...
<input type="checkbox"/>		[Filebeat Suricata] Alert Overview ECS	...
<input type="checkbox"/>		[Filebeat PostgreSQL] Query Duration Overview ECS	...
<input type="checkbox"/>		[Filebeat Zeek] Overview	...
<input type="checkbox"/>		[Filebeat Apache] Access and error logs ECS	...
<input type="checkbox"/>		[Filebeat System] Sudo commands ECS	...

- 一覧にさまざまなグラフ（Visualize）やダッシュボードの名前が表示されます。
6. 一覧のうち1つを選び、Actionsの「…」をクリックし、「Inspect」をクリックします。

Management / Saved objects / Edit dashboard

```
panelsJSON
1 - [
2 - {
3   "panelIndex": "11",
4   "panelRefName": "panel_0",
5   "version": "7.3.0",
6   "gridData": {
7     "x": 0,
8     "y": 16,
9     "w": 48,
10    "h": 12,
11    "i": "11"
12  },
13  "embeddableConfig": {
14    "columns": [
15      "log.level",
16      "message"
17    ],
18    "sort": [

```

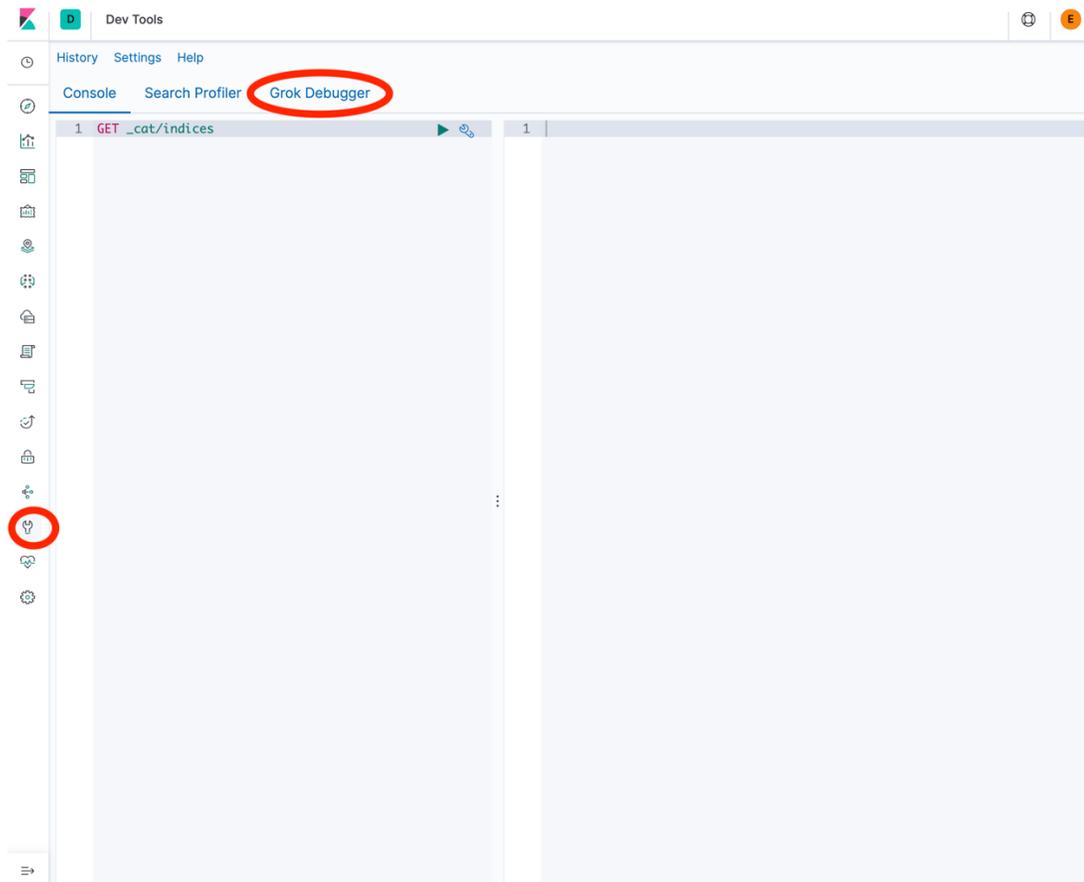
```
references
1 - [
2 - {
3   "name": "panel_0",
4   "type": "search",
5   "id": "9eb25600-a1f0-11e7-928f-5dbe6f6f5519-ecs"
6   },
7 - {
8   "name": "panel_1",
9   "type": "search",
10  "id": "6d9e66d0-a1f0-11e7-928f-5dbe6f6f5519-ecs"
11  },
12 - {
13  "name": "panel_2",
14  "type": "visualization",
15  "id": "1cfb1a80-a1f4-11e7-928f-5dbe6f6f5519-ecs"
16  },
17 - {

```

すると、ダッシュボードやグラフの設定を見ることができます。エディタにて表示されますが、Kibana 自体はこのデータを JSON で扱います。事前にこのように定義されたダッシュボードなどにより、簡単にデータを可視化できるようになっています。

# Grok Debugger で Grok filter を体験

1. Grok Debugger を使い、Grok pattern による文字列の構造化を体験しましょう。Kibana で menu から“Dev Tools” をクリックし、“Grok Debugger”をクリックします。



2. まずはパースする対象となる文字列を入力します。  
"55.3.244.1 GET /index.html 15824 0.043" という文字列を Sample Data に入力します。

Dev Tools

Console Search Profiler Grok Debugger

Sample Data

```
1 55.3.244.1 GET /index.html 15824 0.043
```

Grok Pattern

```
1 %{IP:client} %{WORD:method} %{URIPATHPARAM:request} %{NUMBER:bytes} %{NUMBER:duration}
2
```

> Custom Patterns

Simulate

Structured Data

```
1 - {
2   "duration": "0.043",
3   "request": "/index.html",
4   "method": "GET",
5   "bytes": "15824",
6   "client": "55.3.244.1"
7 }
```

この文字列は、IP アドレス、リクエストの Verb、リクエスト先、バイト数、時間（秒）であるとします。

- 次に、文字列をパースするために、Grok Pattern を記述します。  
まずは、“%{IP:client}”を入力して、Simulate を実行してみましょう。Sample Data の IP アドレスの部分だけが、Structured Data に出力されているのがわかります。では、順番に、“%{WORD:method} %{URIPATHPARAM:request} %{NUMBER:bytes} %{NUMBER:duration}”を入力しながら、Simulate してみましょう。Sample Data 同様に、それぞれの Grok Pattern の間はスペースを入力する必要がありますので注意してください。

Dev Tools

Console Search Profiler Grok Debugger

Sample Data

```
1 55.3.244.1 GET /index.html 15824 0.043
```

Grok Pattern

```
1 %{IP:client} %{WORD:method} %{URIPATHPARAM:request} %{NUMBER:bytes} %{NUMBER:duration}
```

Custom Patterns

Simulate

Structured Data

```
1 - {
2   "duration": "0.043",
3   "request": "/index.html",
4   "method": "GET",
5   "bytes": "15824",
6   "client": "55.3.244.1"
7 }
```

4. 時間のある方は、ご自身がよく扱うログを Sample Data に入力し、Grok Pattern を試してみましょう。定義済みの Grok パターンは以下の URL から確認できます。  
<https://github.com/elastic/elasticsearch/tree/7.4/libs/grok/src/main/resources/patterns>  
<https://github.com/logstash-plugins/logstash-patterns-core/tree/master/patterns>