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はなく、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	https://www.elastic.co/downloads/beats/filebeat

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
```

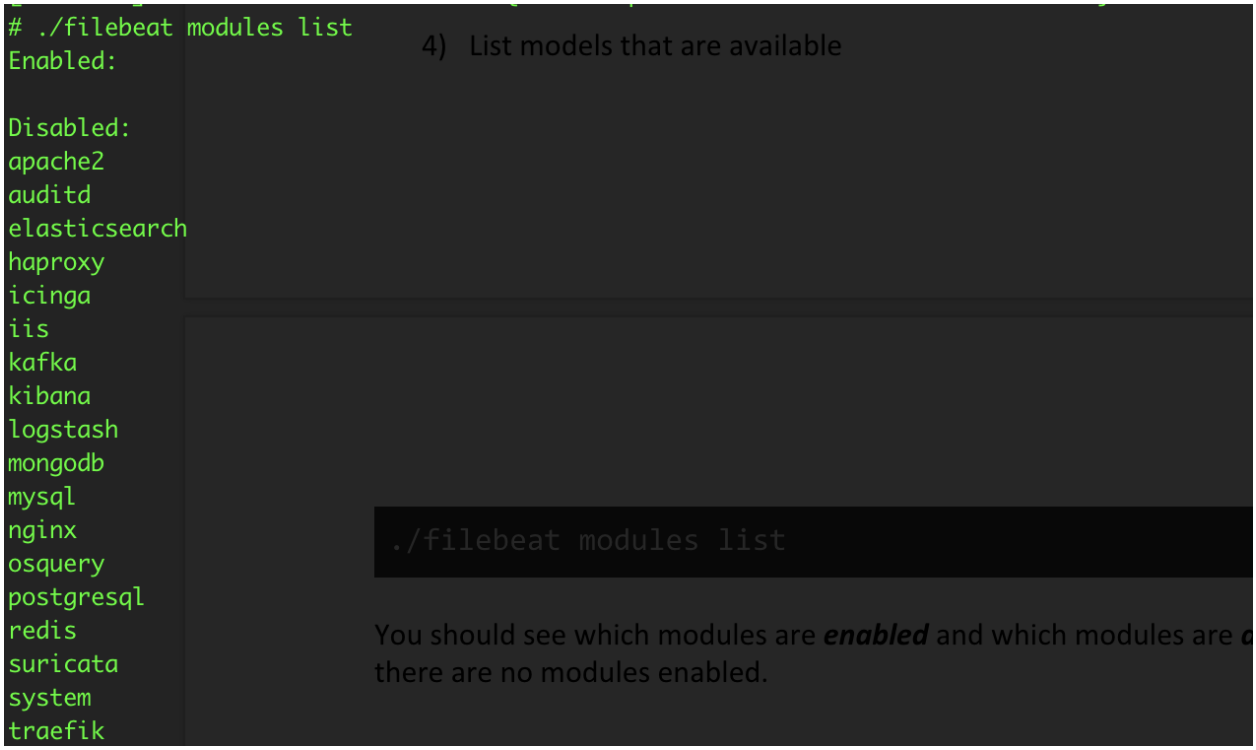
3) filebeat に移動します。

```
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。

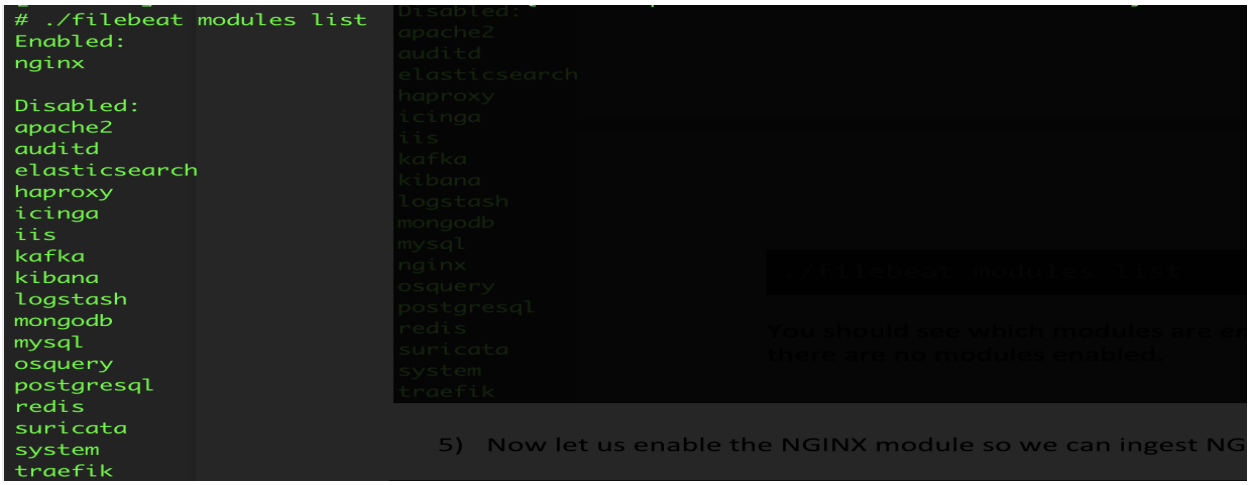
```
./filebeat modules list
```

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。



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

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



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

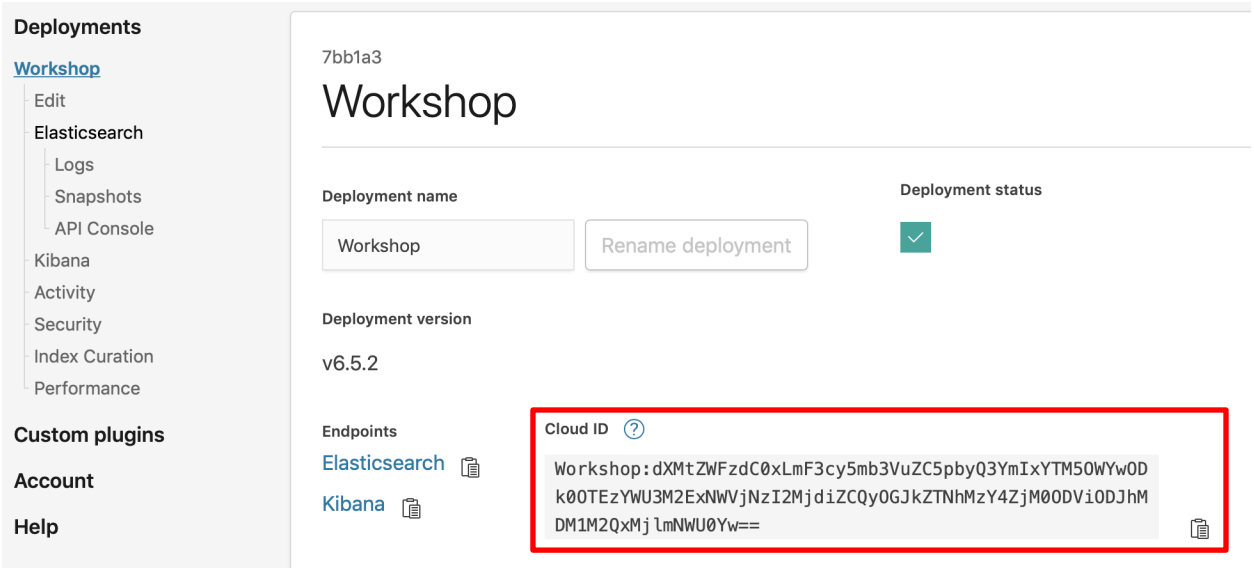
例：(以下は例ですので、実際にはご自身のものをお使いください)

```
#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。




```
#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"
```

7) 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：

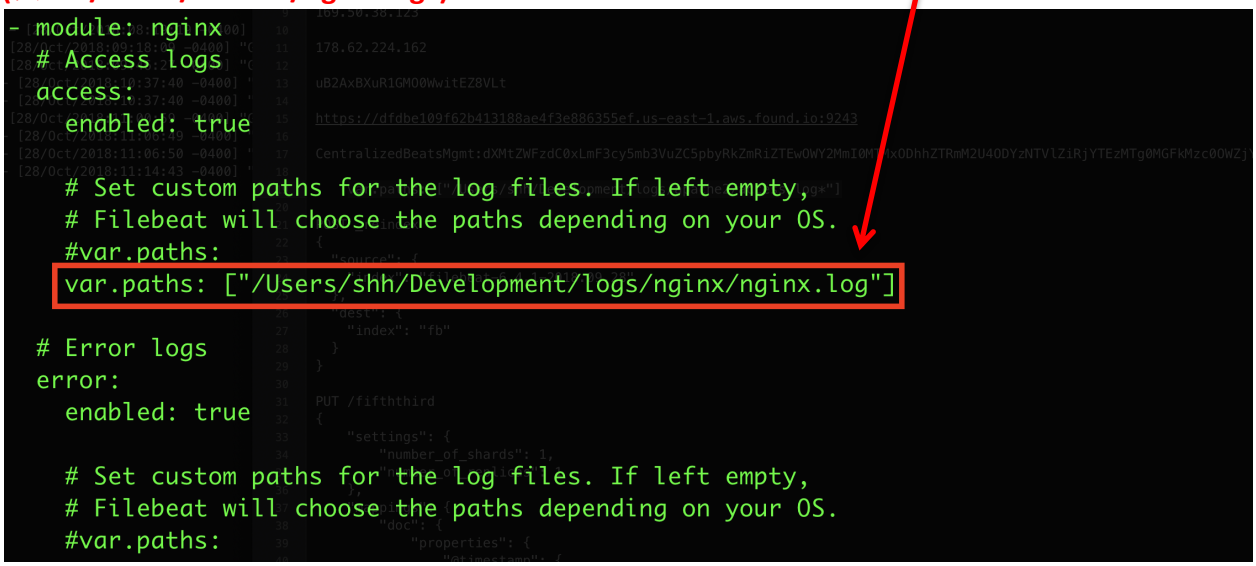
```
wget https://lgworkshop-bucket.s3.amazonaws.com/nginx.log.zip

unzip nginx.log.zip
```

8) 以下のディレクトリに移動します。

```
cd modules.d
```

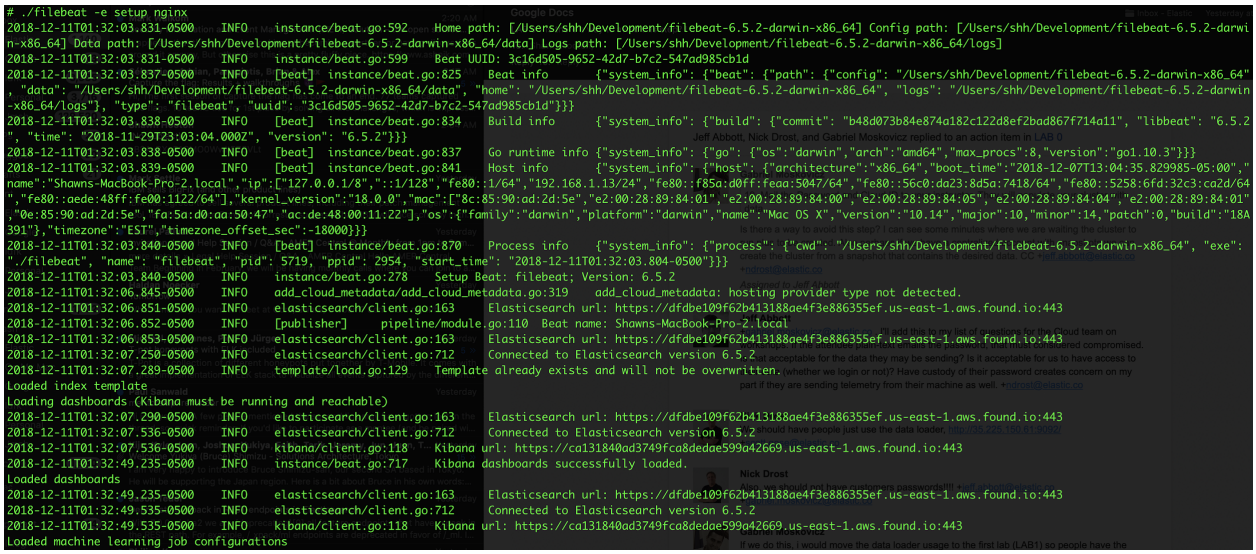
9) NGINX module がダウンロードしたログファイルの場所を参照するようにします。`nginx.yml` ファイルを編集し、ダウンロードしたログファイルのためのエントリを追加します。以下の設定を `nginx.yaml` に追加します。

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")



10) 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。

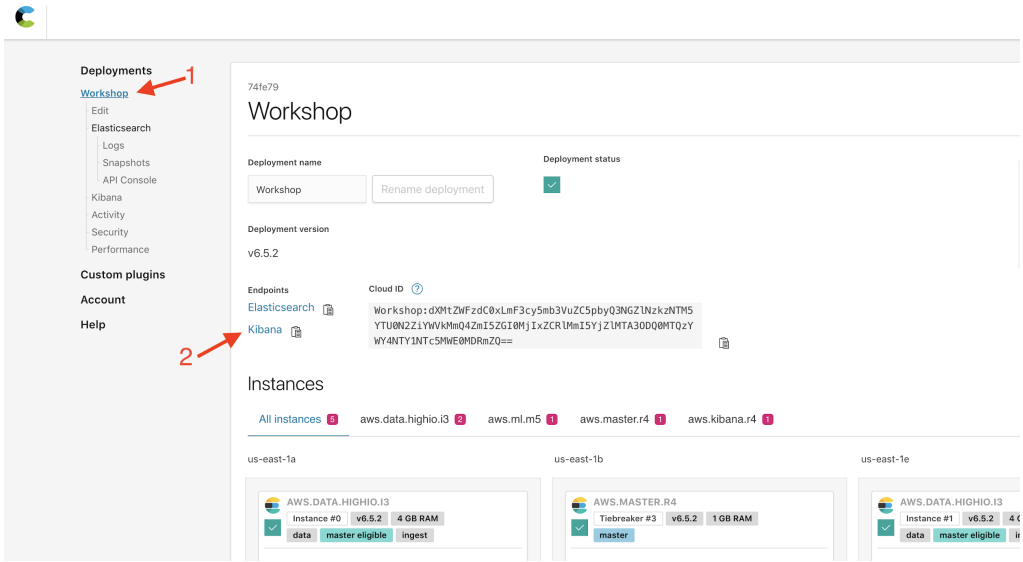
```
cd ..
./filebeat -e setup nginx
./filebeat -e
```



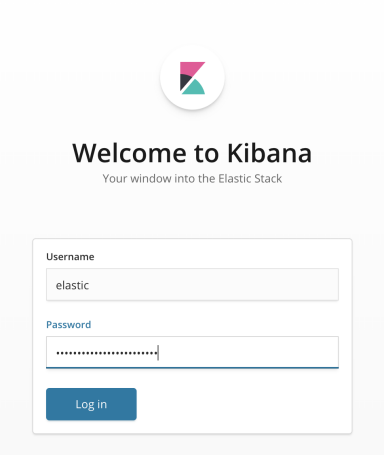
Kibana でデータを確認

Kibana で Index を確認してみましょう。

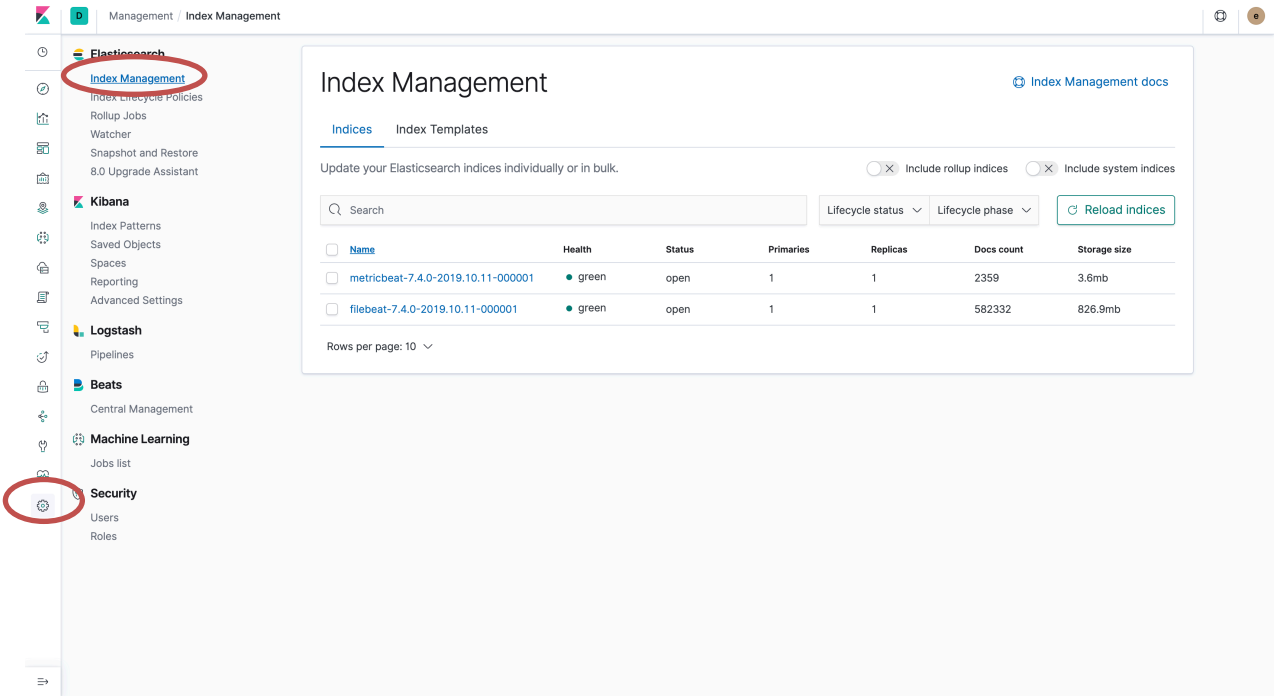
1) クラウドコンソールにログインして、Kibana link をクリックします。



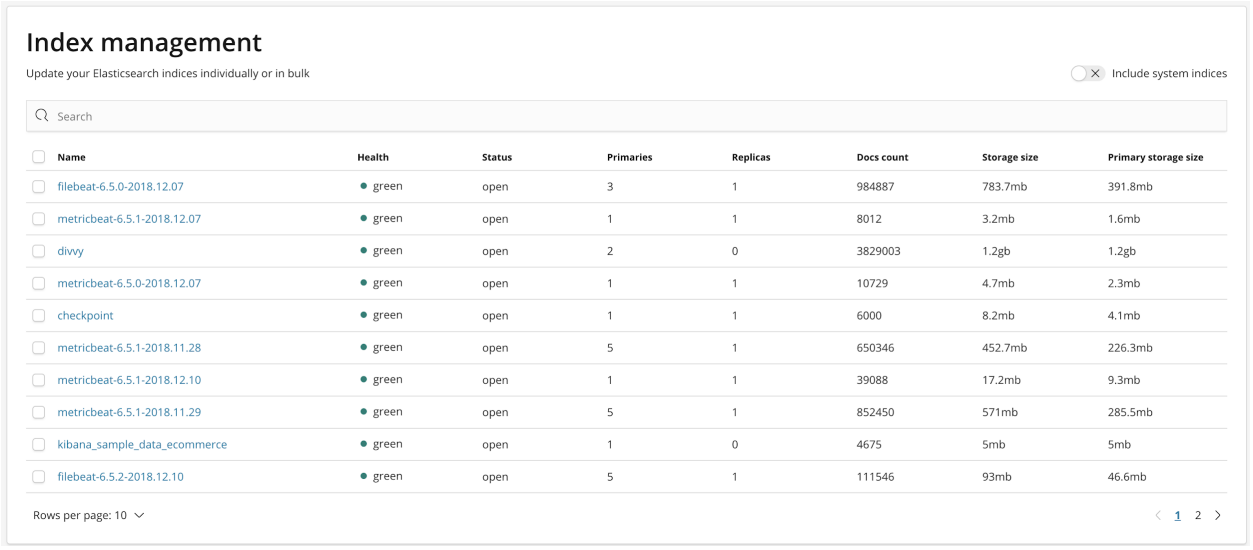
2) Lab 0 で取得したクレデンシャルで Kibana にログインします。



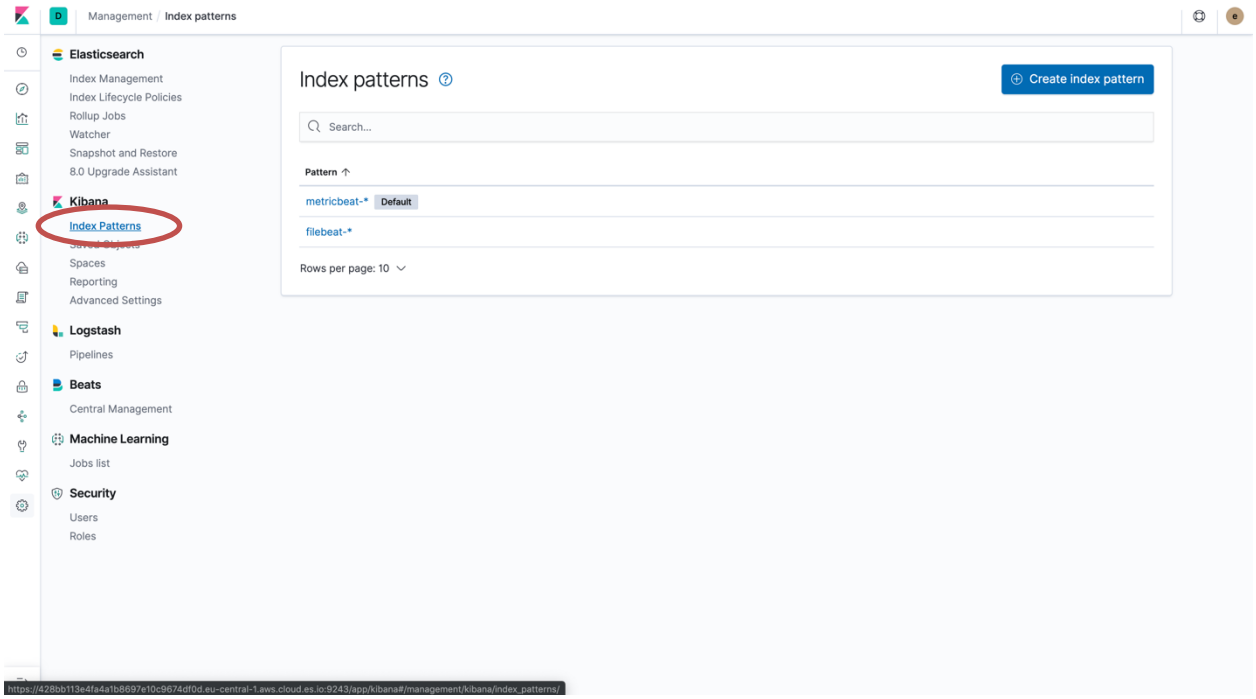
3) Management Link をクリックします。



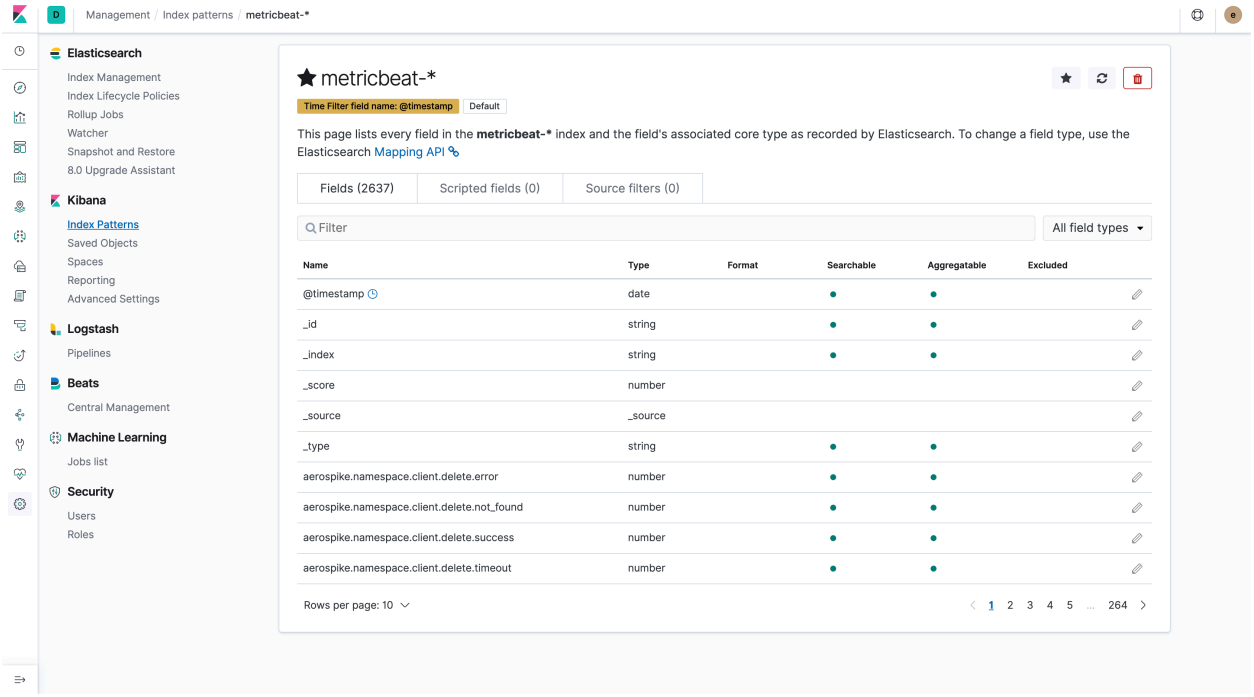
4) filebeat-<version>-YYYY.MM.DD-000001 という Index を探してみてください。version は製品のバージョン、YYYY, MM, DD は年月日を表します。Docs Count, Storage Size も確認してください。



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。

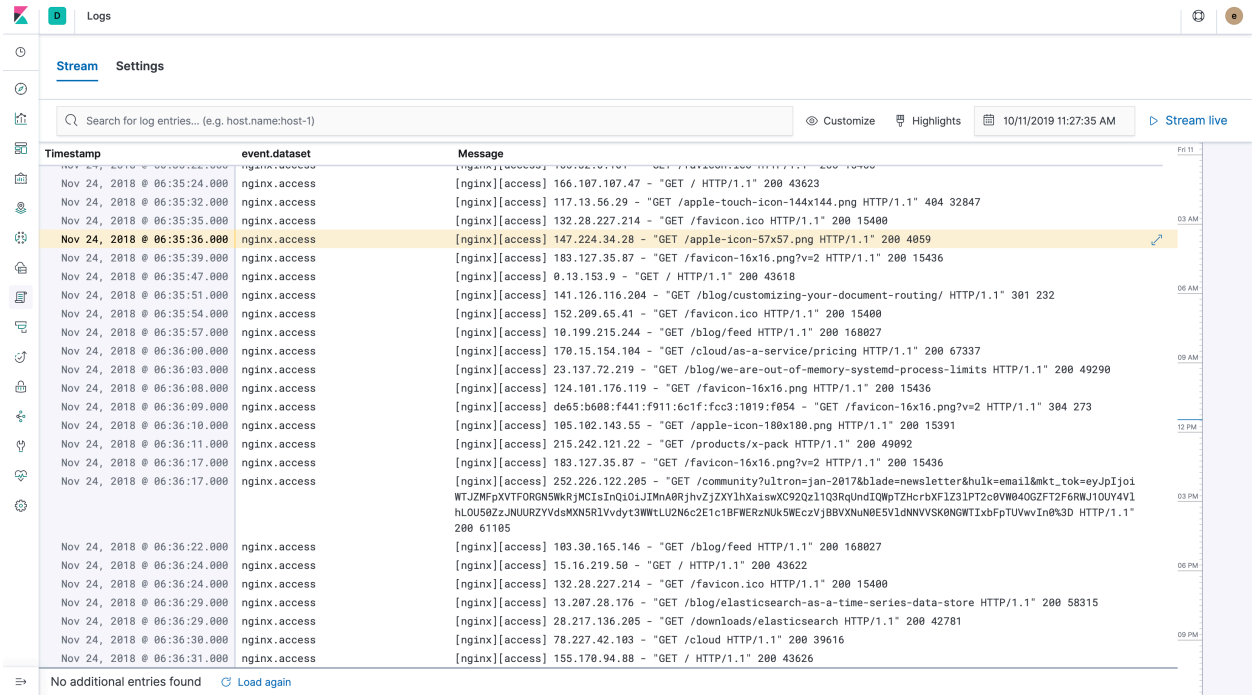


6) filebeat の Index patterns を確認してください。どの field が searchable か、aggregatable かを確認してください。



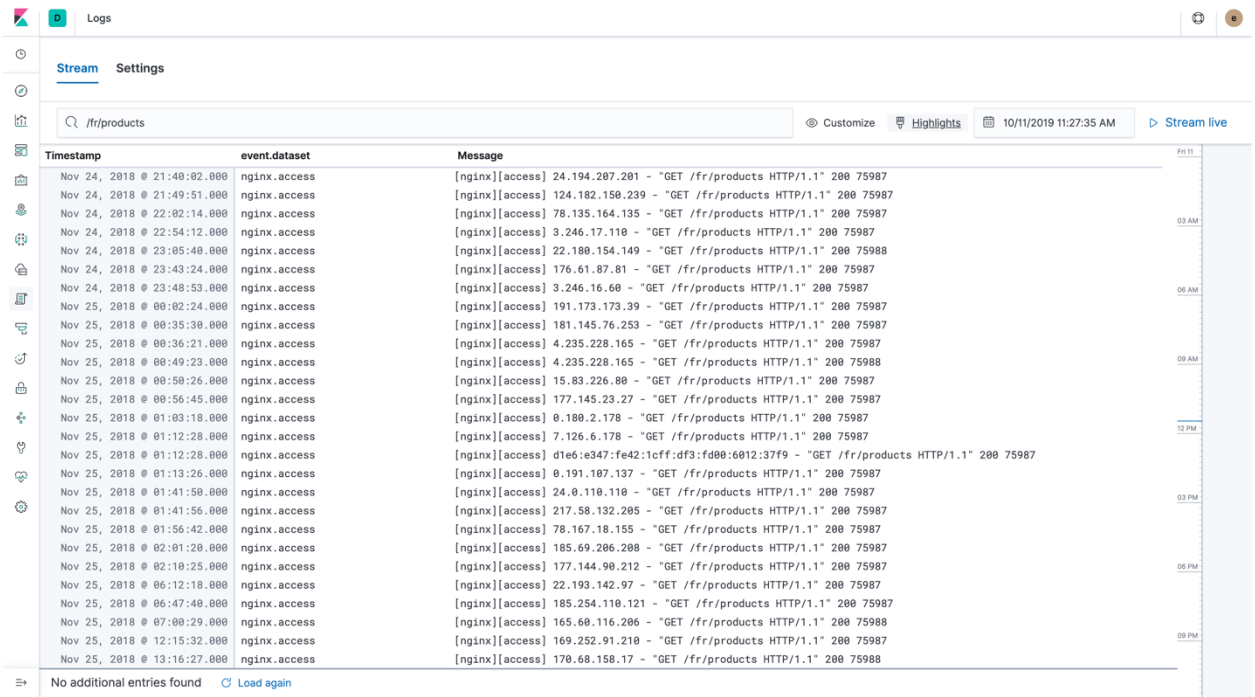
Kibana で filebeat のダッシュボードを体験

1. Kibana で menu から“Logs”をクリックしてみましょう。filebeat-* indices.からのログを見ることができます。



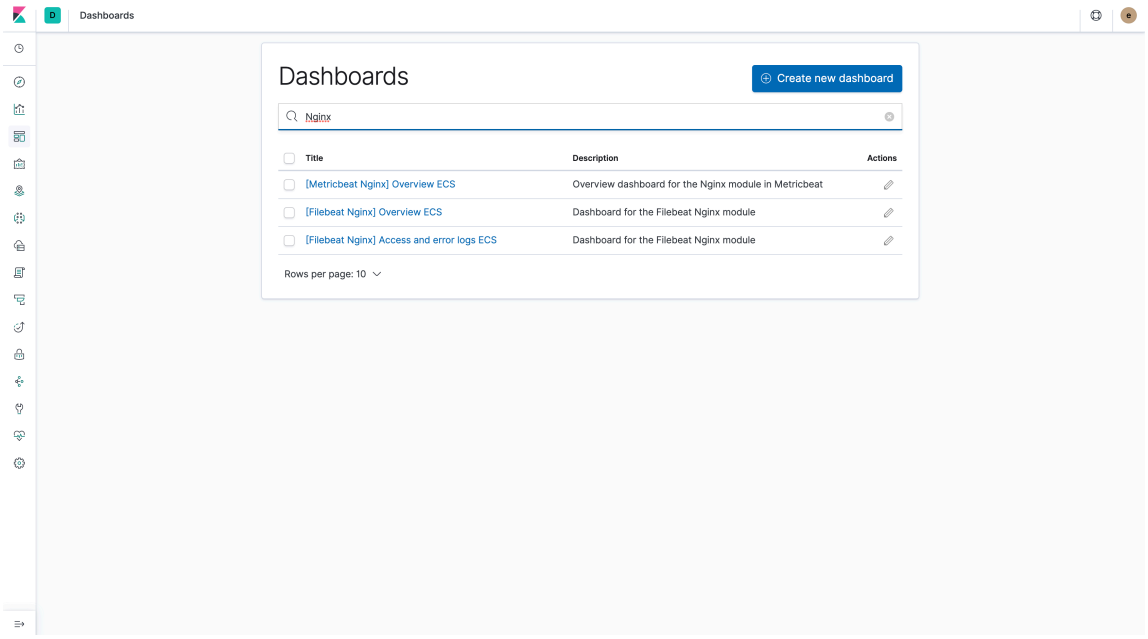
2. 右上の “Stream live”をクリックすると、新しいログが投入されるとスクリーンがそれを表示します。これはログの tail と同様の経験を提供するものです。

3. 検索バーから“`/fr/products`“を検索してみます。

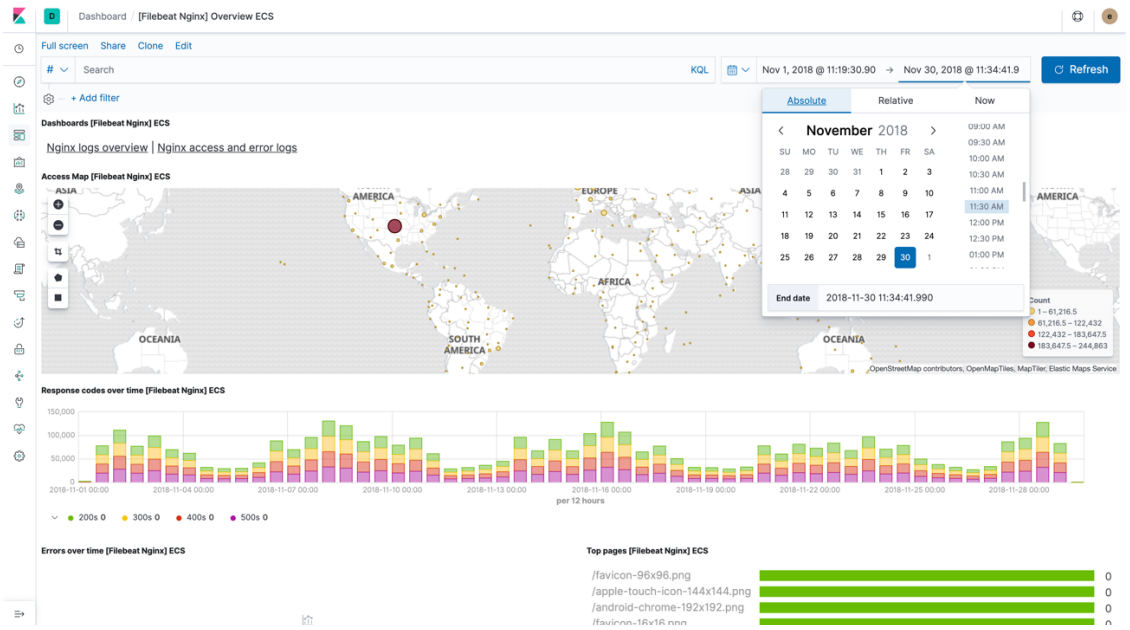


この機能は、検索エンジン(Elasticsearch)によるものです。検索の機能は、ロギングでも非常に有効です。

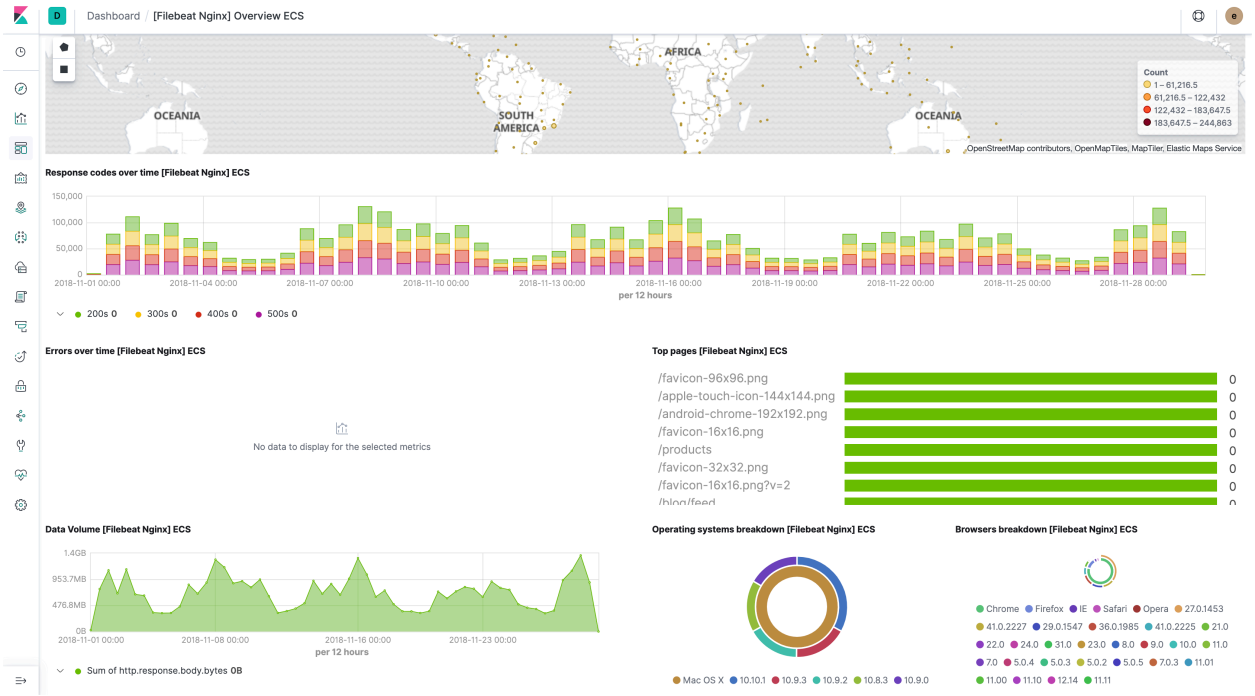
4. それでは、Filebeat のアウトオブボックスの Dashboards を見てみましょう。Menu の Dashboards をクリックします。全ての Dashboards のリストが表示されます。 “Nginx”を検索してみます。“[Filebeat Nginx] Overview ECS” をクリックします。



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

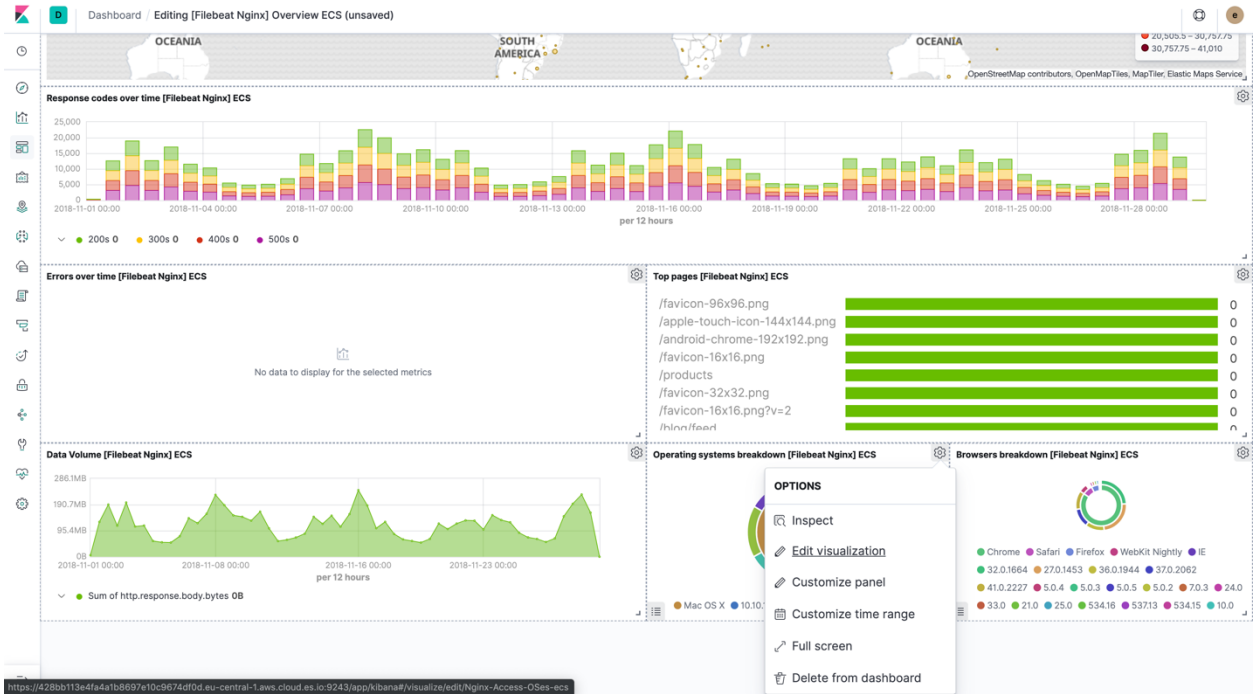


以下のような Dashboard が表示されるはずです。

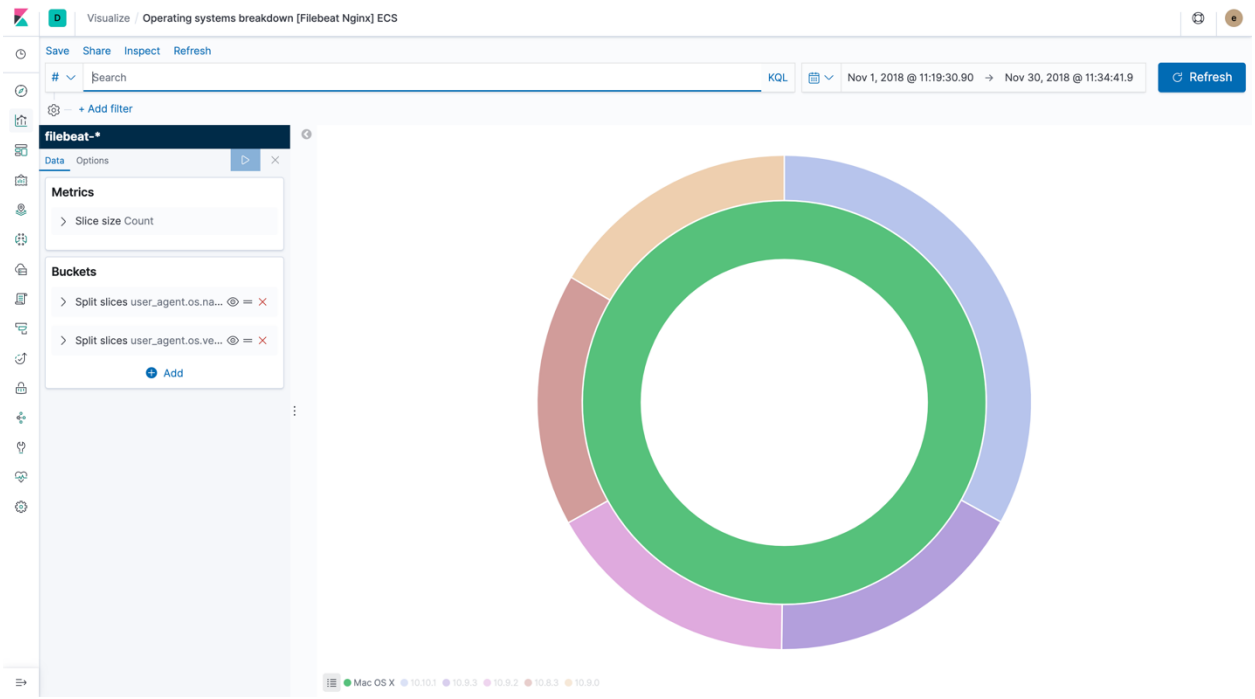


November 28th までのデータが全て見えない場合、まだデータがローディング中かも知れません。Auto-Refresh (next to date picker) を On にして dashboard がリアルタイムに更新されるのを見てみましょう。

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。



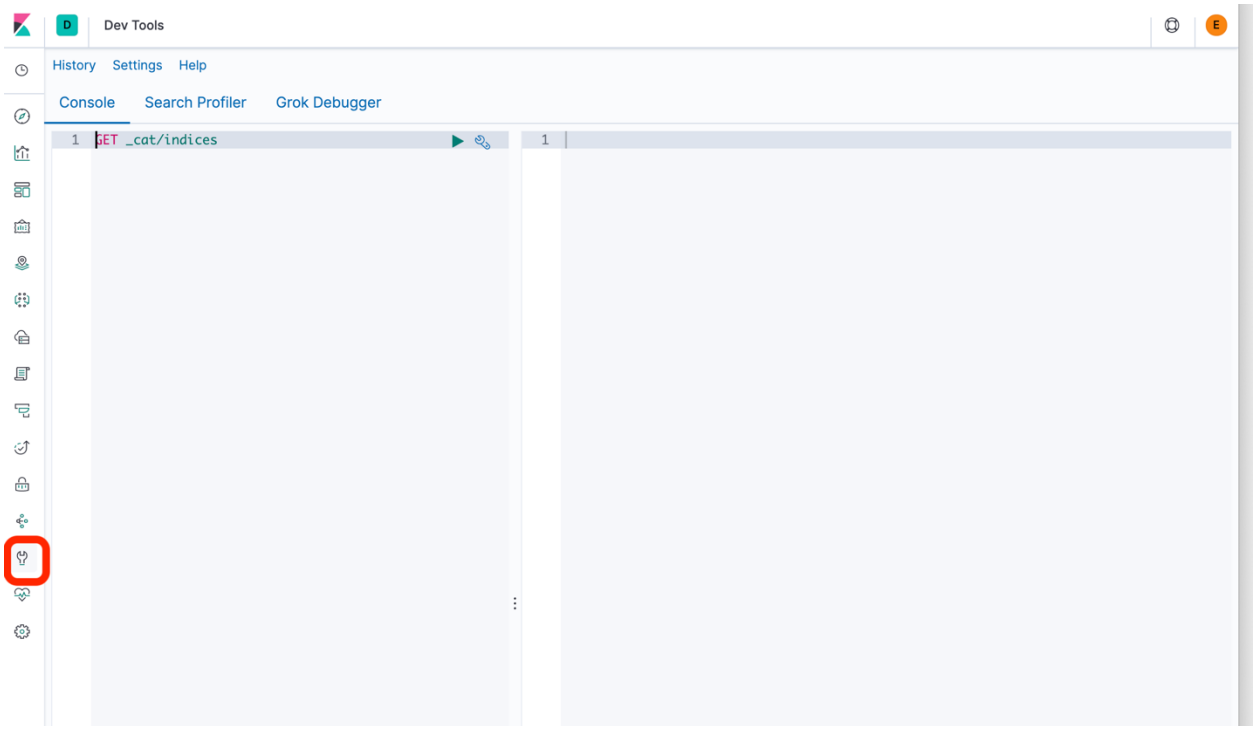
Pie chart visualization の例：


Visualize / Operating systems breakdown [Filebeat Nginx] ECS
Save
Share
Inspect
Refresh
Search
KQL
Nov 1, 2018 @ 11:19:30.90
Nov 30, 2018 @ 11:34:41.9
Refresh
+ Add filter
filebeat-*
Data
Options
Metrics
Slice size Count
Buckets
Split slices user_agent.os.na...
Split slices user_agent.os.ve...
Add
Mac OS X

filebeat のモジュールの仕組みを確認

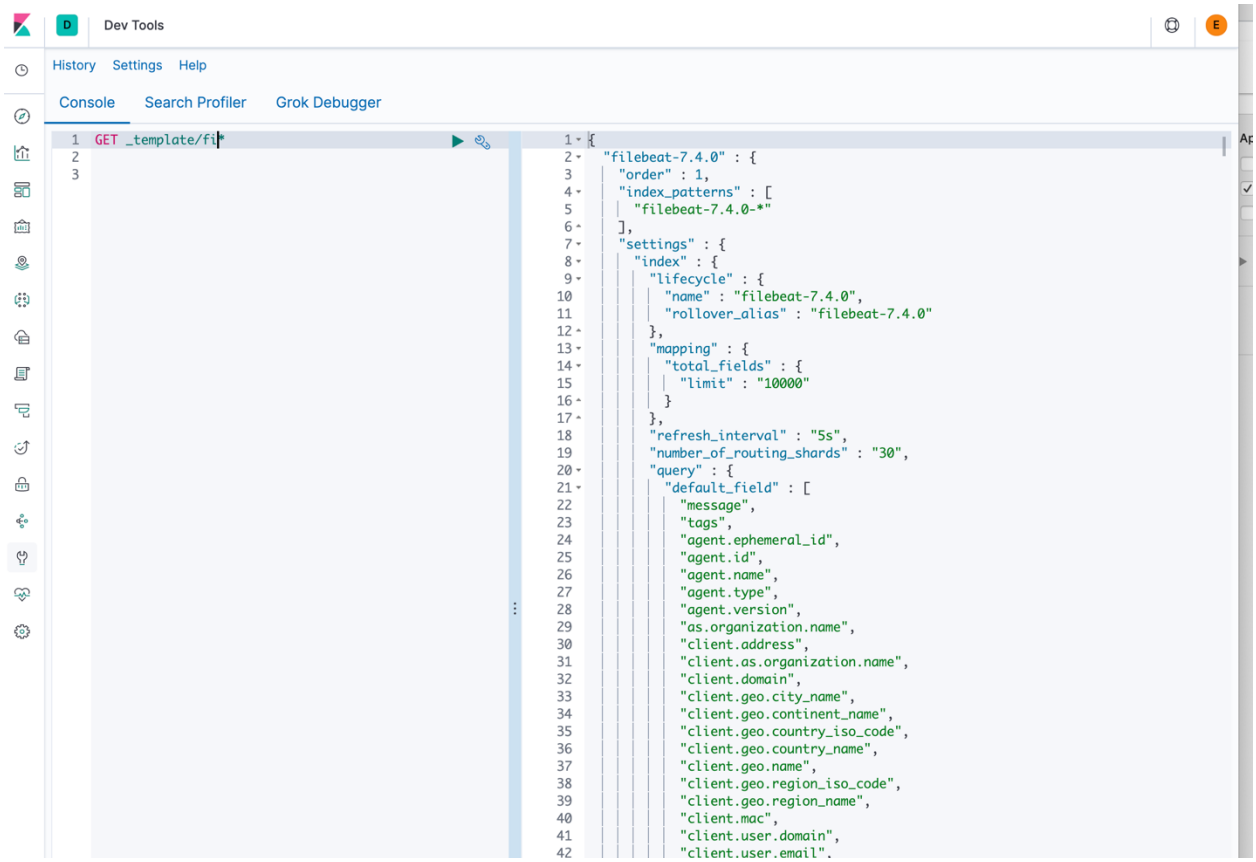
Filebeat のモジュールがどのような仕組みでログの文字列を構造化かしているのかを確認しましょう。

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。

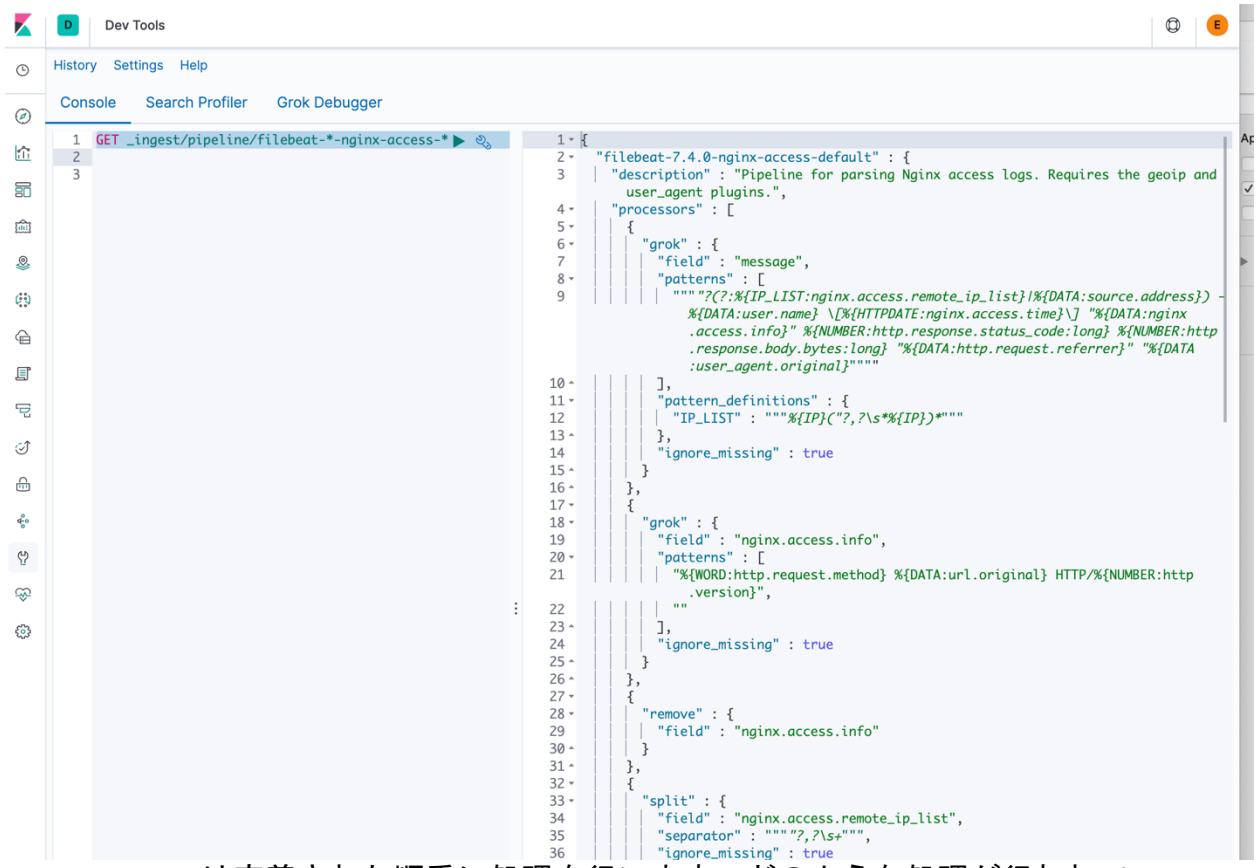


2. Index Template の確認のために、”GET _template/fi*”と入力して、右側の緑の三角（再生）ボタンをクリックします。すると、右側に Elasticsearch からのレスポン

スが帰ってきます。

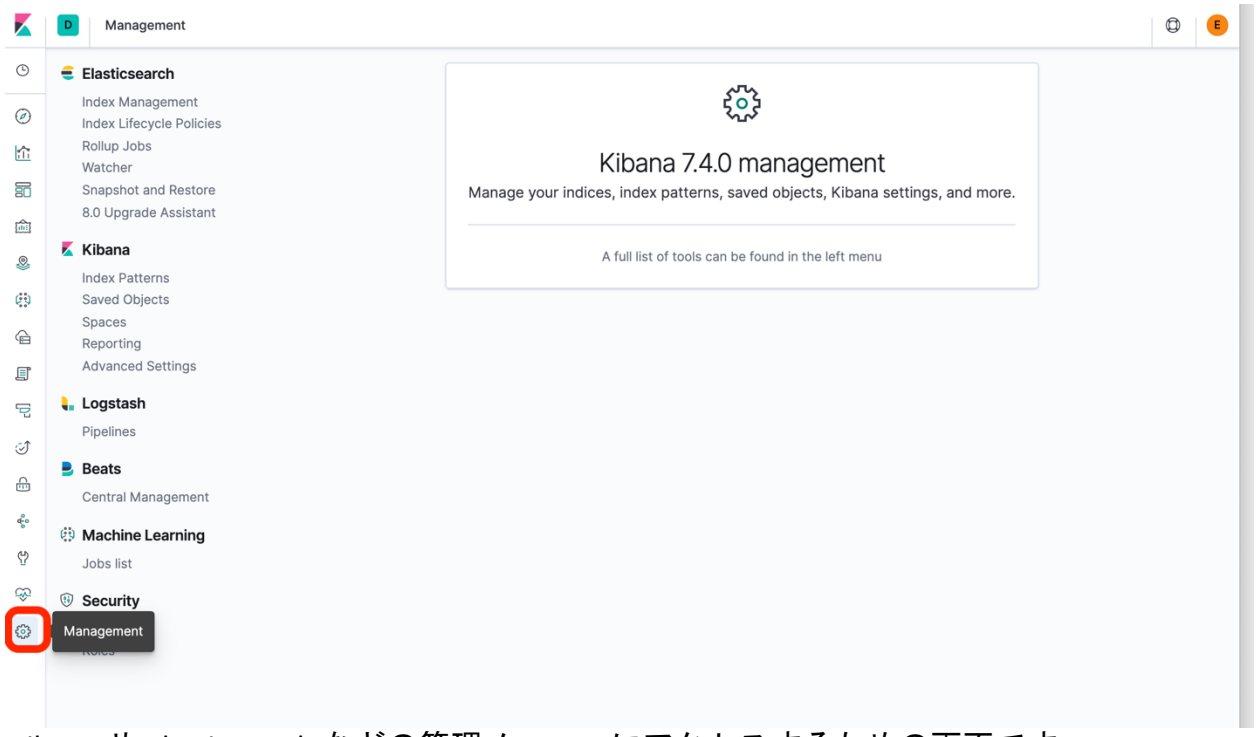


3. 次に Ingest Pipeline を確認しましょう。先ほど NGINX のログを Filebeat で取り込む設定をしました。Filebeat が Elasticsearch に送信した 1 行のログ文字列を、Elasticsearch がデータを取り込むタイミングでどのような処理を行っているかが、Ingest Pipeline を確認することでわかります。Dev Tools の Console で” GET _ingest/pipeline/filebeat-*-nginx-access-*”と入力して、実行しましょう。



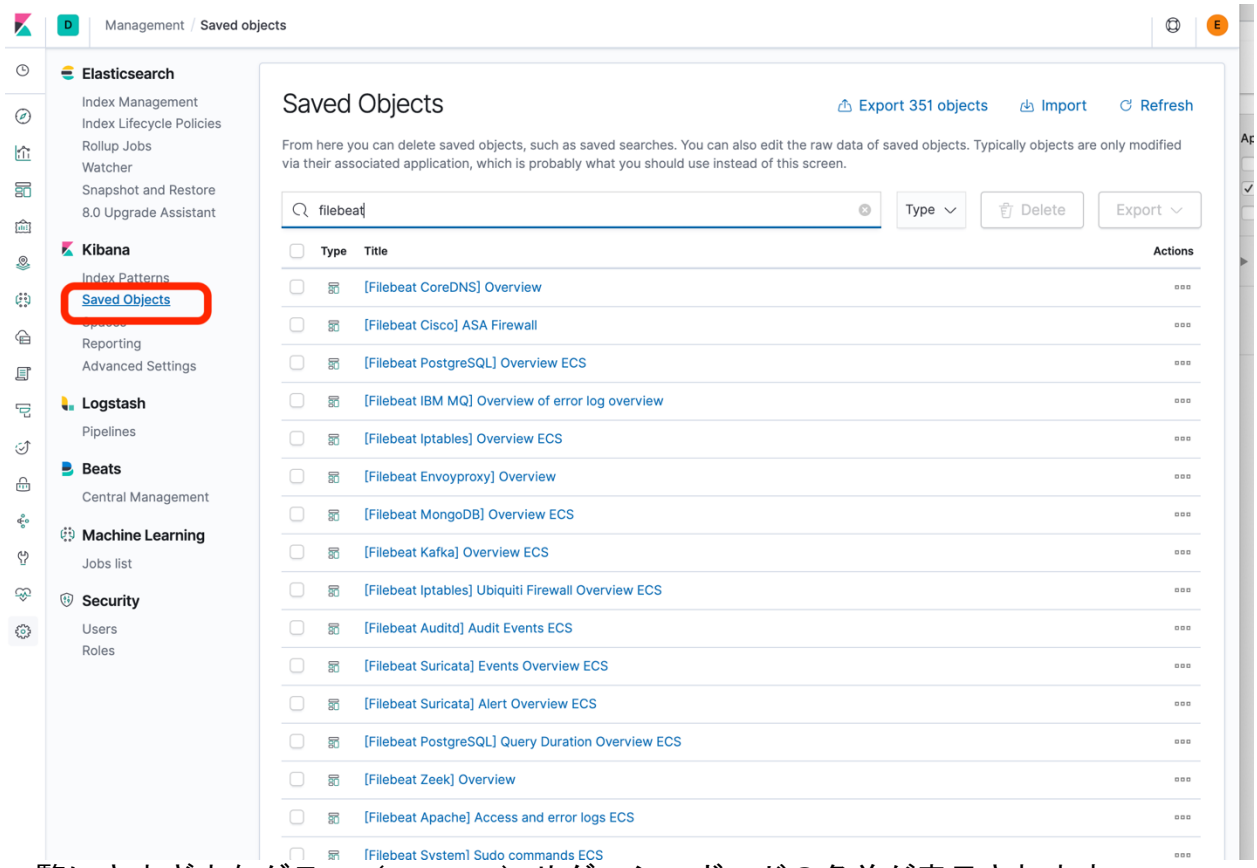
Ingest Pipeline は定義された順番に処理を行います。どのような処理が行われているかを JSON で確認してみましょう。

4. 最後にグラフとダッシュボードの設定である Saved Objects を確認しましょう。まずは左のメニューから”Management”をクリックします。



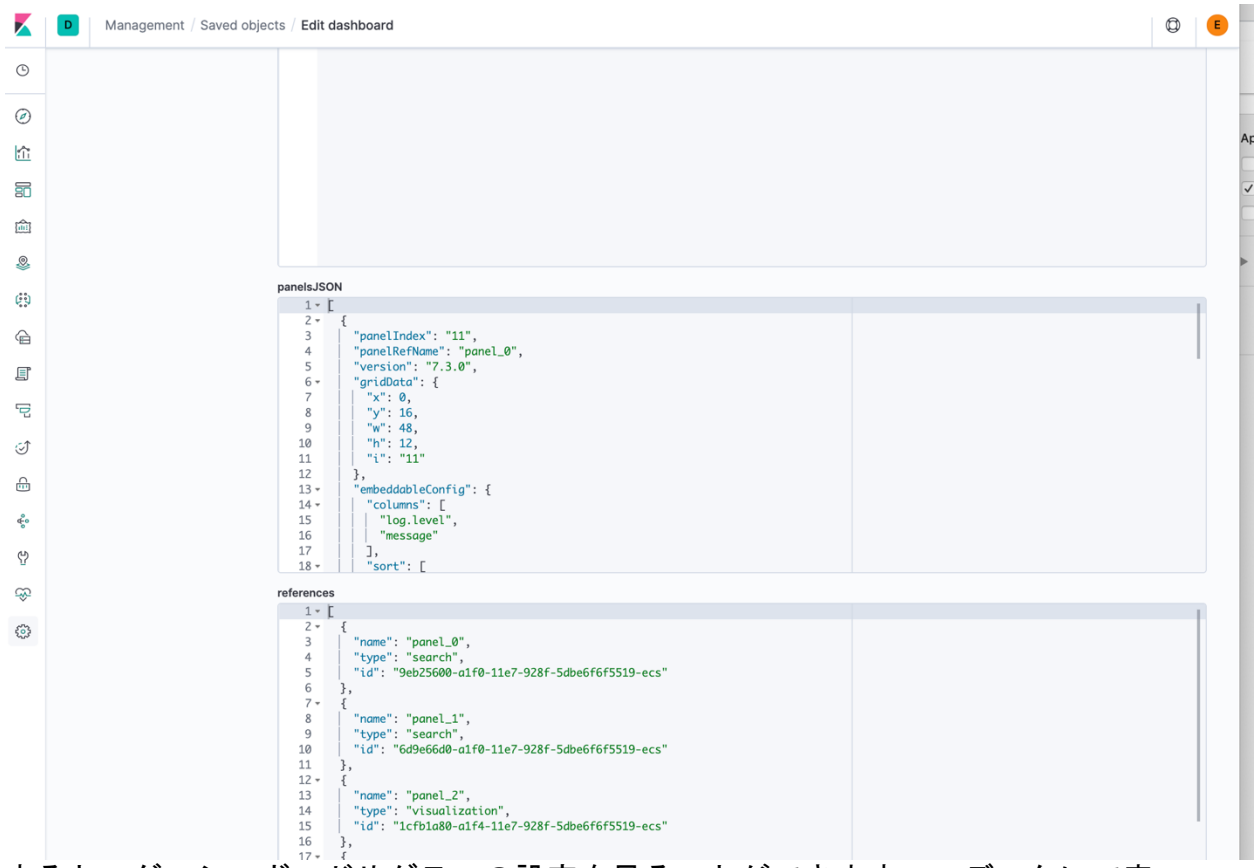
Kibana や Elasticsearch などの管理メニューにアクセスするための画面です。

5. 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。



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。

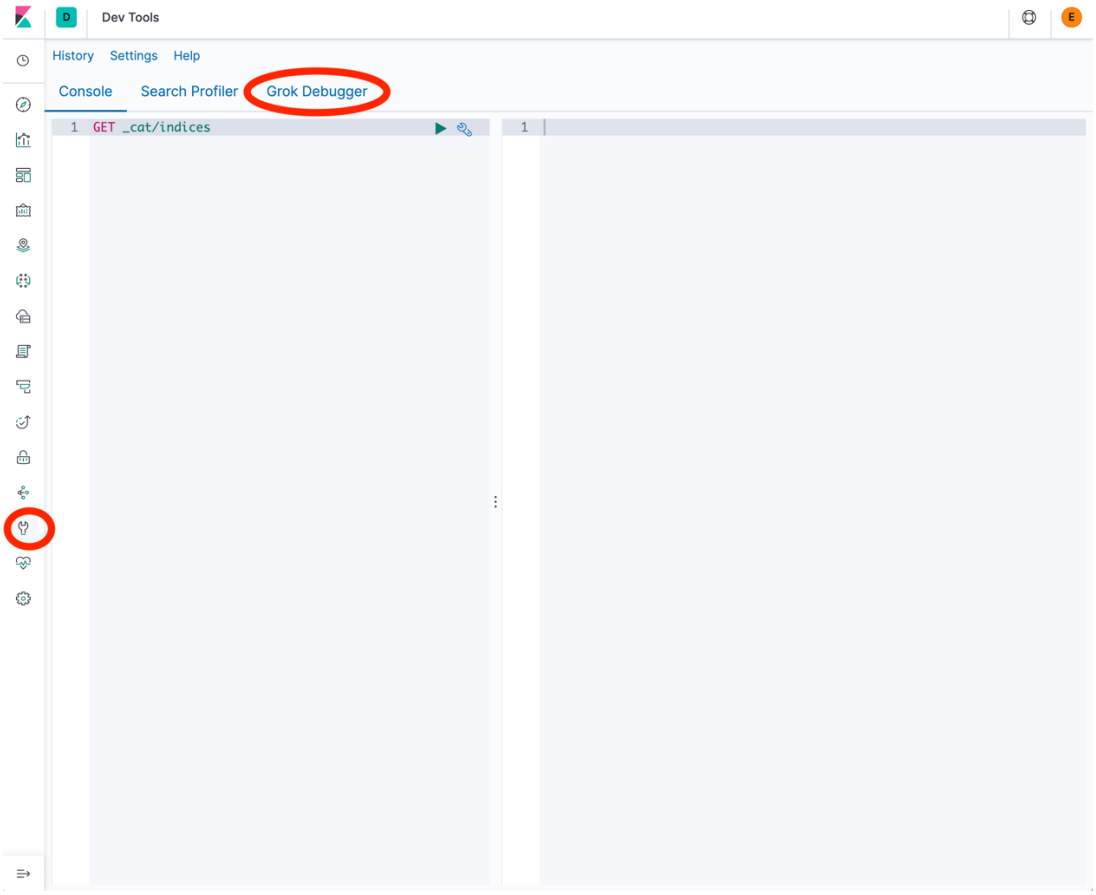
6. 一覧のうち 1 つを選び、Actions の「…」をクリックし、”Inspect”をクリックします。



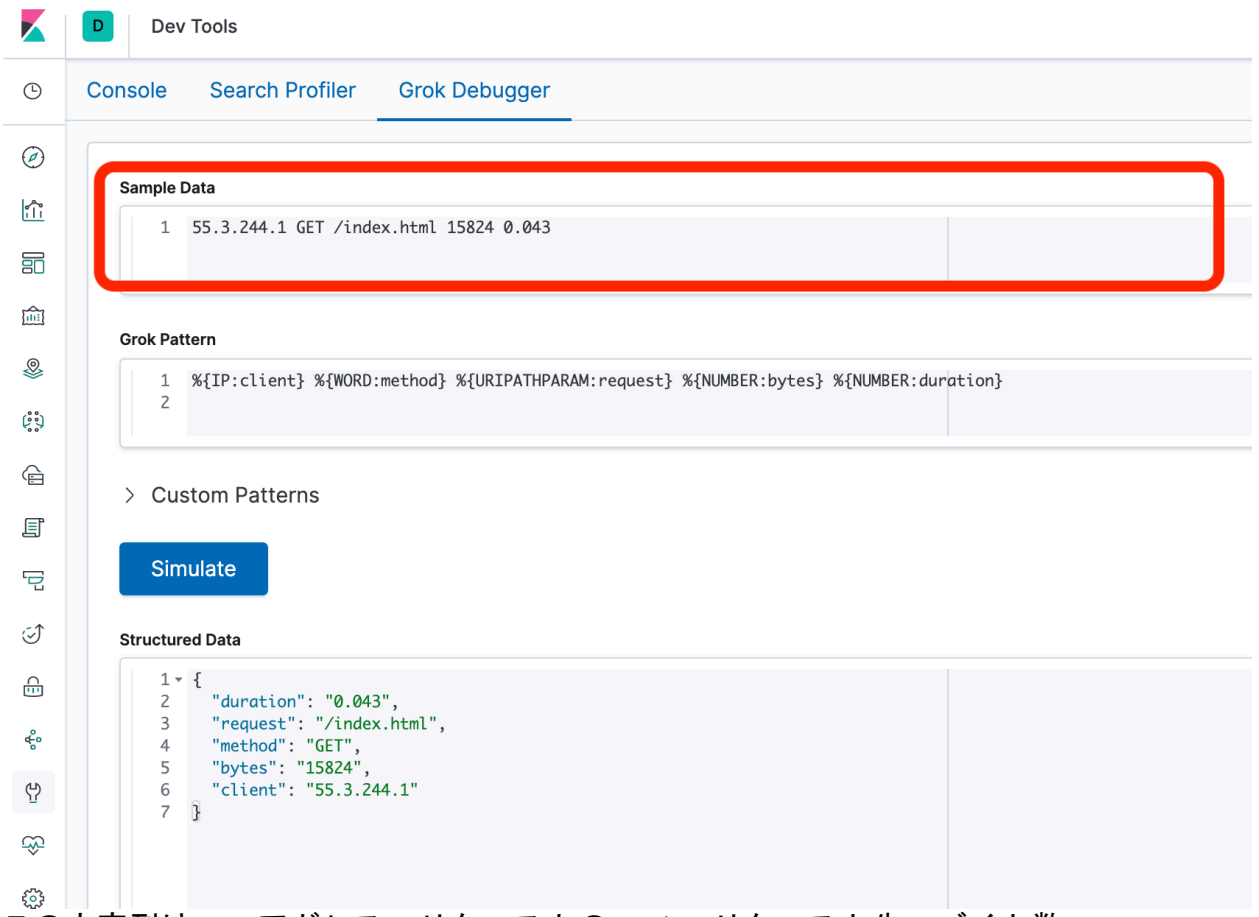
すると、ダッシュボードやグラフの設定を見ることができます。エディタにて表示されますが、Kibana 自体はこのデータを JSON で扱います。事前にこのように定義されたダッシュボードなどにより、簡単にデータを可視化できるようになっています。

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。

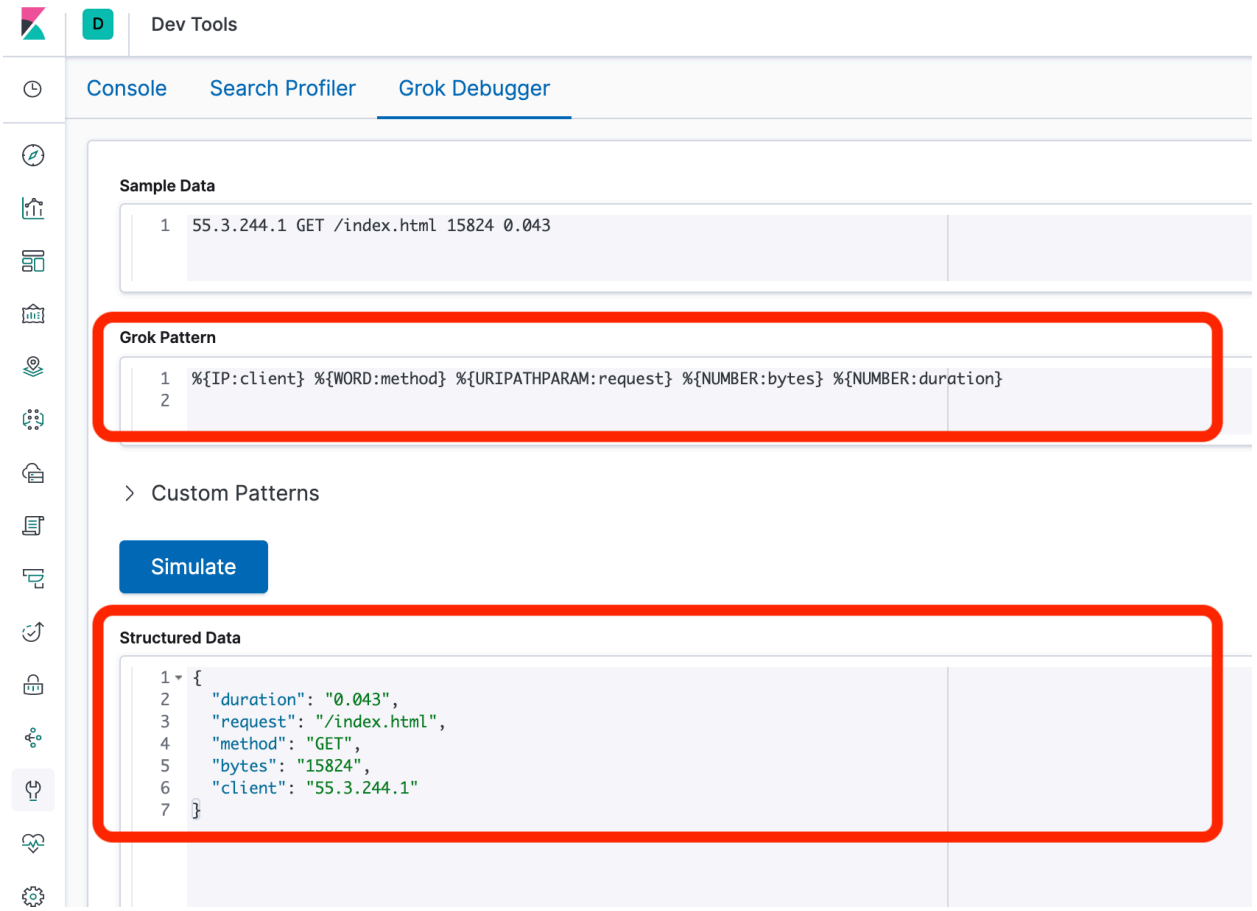


2. まずはパースする対象となる文字列を入力します。
”55.3.244.1 GET /index.html 15824 0.043” という文字列を Sample Data に入力します。



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。

3. 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るので注意してください。



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