Не так давно я рассказывал о том, как настроить ELK Stack для централизованного хранения логов. Сегодня хочу подробно рассказать о том, как создать координатную географическую карту на основе логов nginx и составить дашборд для него же. На этом дашборде очень удобно мониторить состояние веб проекта — расследовать иниценденты, анализировать ошибки.



Содержание:

- 1 Введение
- 2 Создание index шаблона
- 3 Настройка координатной карты в Kibana
- 4 Настройка Dashboard для nginx
- 5 Заключение

## Введение

Начнем создание дашборда с самого сложного — настройки гео карты запросов. На официальном сайте есть подробный мануал на тему создания GeoIP карты. В нем вроде бы все понятно. Никаких особых настроек не требуется. Все работает из коробки. Но у меня никак не хотело работать все то, что там описано. Пришлось прилично поковыряться с elasticsearch и его шаблонами, чтобы разобраться в чем причина.

Все дело в том, что описанный в инструкции способ работает из коробки, только если вы используете стандартный шаблон для индексов в формате logstash-*. Скорее всего у вас будет много разных шаблонов и индексов после того, как вы запустите систему в промышленную эксплуатацию.

Основная сложность тут в том, что для работы geoip карты вам нужны в шаблоне поля с типом **geo_point**. После создания индекса, тип полей уже нельзя поменять. То есть просто преобразовать данные на основе ip в координаты не сложно, это умеет делать модуль geoip в logstash. Но вот дальше вы никак не превратите координаты в виде числа в geo_point данные. Нужно в самом начале создать шаблон с такими полями.

Надеюсь понятно объяснил :) Если не понятно сразу, то сообразите дальше по ходу моего рассказа. Я сам пока разобрался в этой кухне, прилично поковырялся и нагуглился.

В дальнейшем я буду считать, что ваш elasticsearch и kibana настроены примерно как у меня в инструкции. Фильтр logstash, отвечающий за обработку логов nginx выглядит следующим образом:

```
if [type] == "nginx-ext-access" {
        grok {
            match => [ "message" , "%{COMBINEDAPACHELOG}+%{GREEDYDATA:extra_fields}"]
            overwrite => [ "message" ]
            }
        mutate {
            convert => ["response", "integer"]
            convert => ["bytes", "integer"]
            convert => ["responsetime", "float"]
            }
        geoip {
            source => "clientip"
            target => "geoip"
            add_tag => [ "nginx-geoip" ]
            }
        date {
            match => [ "timestamp" , "dd/MMM/YYYY:HH:mm:ss Z" ]
            remove_field => [ "timestamp" ]
            }
        useragent {
            source => "agent"
            }
```

```
    }
```

И вот так логи уходят в elasticsearch

```
if [type] == "nginx-ext-access" {
        elasticsearch {
            hosts     => "localhost:9200"
            index    => "nginx-ext-%{+YYYY.MM.dd}"
        }
    }
```

## Создание index шаблона

Как я уже сказал выше, для того, чтобы у вас заработала geoip карта, у вас должны быть в шаблоне индекса поля типа **geo_point**. Если их не будет, то вы сразу при создании визуализации с Coordinate Map получите ошибку:

```
No Compatible Fields: The "nginx-*" index pattern does not contain any of the following field types: geo_point
```

Что я только не делал, после того, как получил эту ошибку. Я проверял работу geoip модуля. Смотрел поля с координатами на основе ip адреса. Все было в порядке и все было на месте.

Но geoip карта в Kibana не работала. Поуглив немного эту тему, я потихоньку стал понимать, в чем тут дело.

Для начала посмотрим шаблон нашего индекса с логами nginx. Для этого идем в **Management -> Index Management**. Выбираем наш индекс и смотрим **Mapping**. Нас интересует поле **location**.

Оно имеет тип **float**, а нам нужно, судя по статье на сайте, тип **geo_point**.

Дальше стал разбираться, как изменить тип поля в шаблоне. Оказалось, что это сделать нельзя. Тип полей можно установить только в момент создания индекса из шаблона. Значит нужно понять, как сделать свой шаблон с нужными полями.

Для начала посмотрим, какие шаблоны у нас сейчас установлены. Для этого идем в **Dev Tools** и выполняем команду:

```
GET /_template
```

Обращаем внимание на шаблон logstash. В нем есть все, что нам нужно. Если ваш индекс будет иметь шаблон logstash-*, то вам ничего настраивать не надо, все заработает из коробки. Мы же добавим новый шаблон nginx* и установим у него параметры полей, необходимые для работы geoip карты.

Выполняем следующий код для создания шаблона nginx по аналогии с шаблоном logstash.

```
PUT _template/nginx
{
"index_patterns": [
      "nginx*"
    ],
    "settings": {
      "index": {
        "refresh_interval": "5s"
      }
    },
    "mappings": {
      "_default_": {
```

```
      "dynamic_templates": [
        {
          "message_field": {
            "path_match": "message",
            "match_mapping_type": "string",
            "mapping": {
              "type": "text",
              "norms": false
            }
          }
        },
        {
          "string_fields": {
            "match": "*",
            "match_mapping_type": "string",
            "mapping": {
              "type": "text",
              "norms": false,
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            }
          }
        }
      ],
      "properties": {
        "@timestamp": {
          "type": "date"
        },
        "@version": {
```

```
          "type": "keyword"
        },
        "geoip": {
          "dynamic": true,
          "properties": {
            "ip": {
              "type": "ip"
            },
            "location": {
              "type": "geo_point"
            },
            "latitude": {
              "type": "half_float"
            },
            "longitude": {
              "type": "half_float"
            }
          }
        }
      }
    },
    "aliases": {}
}
```

Проверяем список доступных шаблонов.

Все в порядке. Теперь новые индексы, попадающие под этот шаблон, будут содержать необходимые поля. Можете либо удалить текущие индексы, для создания новых, либо подождать, когда они сами создадутся в соответствии с вашими правилами.

Прежде чем двигаться дальше, проверьте, что у вас в шаблоне индекса действительно есть поле geo_point. Идем в **Management -> Index Patterns** и смотрим поля нашего индекса, предварительно обновив их, нажав **Refresh field list**.

Если у вас так же, можно двигаться дальше.

На всякий случай расскажу про неправильный путь, по которому я пошел изначально, пытаясь решить проблему с шаблоном. Я узнал, что logstash хранит свои шаблоны в директории */usr/share/logstash/vendor/bundle/jruby/2.3.0/gems/logstash-output-elasticsearch-9.2.0-java/lib/logstash/outputs/elasticsearch* (пипец какой путь :)). Я решил изменить его шаблон, для этого просто отредактировал файл *elasticsearch-template-es6x.json*, поменяв шаблон для индекса. Перезапустил logstash, но ничего не изменилось. Этот шаблон был залит в elasticsearch при первом запуске. Потом уже не меняется. Его надо удалить, чтобы он установился заново с моими изменениями. Я не стал это делать, а просто загрузил новый шаблон.

## Настройка координатной карты в Kibana

Теперь создадим географическую карту с распределением запросов nginx по этой карте на основе ip адресов. Идем в раздел **Visualize** и добавляем **Coordinate Map**. Выбираем индекс с логами nginx. Указываем в карте поле с координатами — **geoip.location**.

Запускаете визуализацию и смотрите результат.

Теперь эту карту можно добавить на дашборд вместе с остальными графиками. Не буду рассказывать, как добавить обычные графики. Там хоть и не совсем все очевидно, но не так сложно. Лучше самим разобраться и порисовать различные графики, чтобы понять, какая визуализация для вас наиболее удобна. Я подобрал по своему вкусу. Много раз перерисовывал и переделывал, пока не удовлетворился результатом.

## Настройка Dashboard для nginx

Я настроил вот такой дашборд в Kibana для логов Nginx (кликабельно, большая картинка, откройте в отдельной вкладке, чтобы рассмотреть).

Здесь представлена следующая информация:

1. Geoip карта
2. Распределение запросов по странам.
3. Список самых популярных урлов.
4. Список самых активных IP.
5. Распределение запросов по типам ответов.
6. Траффик.
7. Непосредственно логи nginx в чистом виде.

С таким дашбордом очень удобно расследовать инциденты и просто смотреть статистику. Выбираем, к примеру, код ошибки и смотрим всю информацию по нему. Сразу подсвечиваются ip, которые спамят запросы. По ним тут же можно получить всю информацию — откуда они и по каким урлам спамят. И так далее. В общем, очень удобно. Я уже не представляю большой веб проект без такого дашборда. Раньше анализ логов для меня был гораздо сложнее. И как я раньше админил без такого инструмента :) Век живи — век учись.

## Заключение

> Не понравилась статья и хочешь научить меня администрировать? Пожалуйста, я люблю учиться. Комментарии в твоем распоряжении. Расскажи, как сделать правильно!

Я долго размышлял над дашбордом для логов nginx. Рисовал графики, выбирал данные. В итоге остановился на таком варианте. Больше ничего полезного для вывода придумать не смог. Кстати, сама Geo карта тут больше для красоты. Я ей не пользуюсь. Практической пользы в ней не вижу. Если у вас есть советы по каким-то еще полезным данным, которые можно вывести — делитесь. Конечно, можно добавить инфу по юзерагентам, системам и браузерам. Но мне кажется, такие вещи удобнее смотреть в сторонней аналитике. Там будут более точные данные.

Отдельно стоит добавить в логи nginx информацию о request_time, upstream_response_time, upstream_cache_status и т.д. Потом эту информацию распарсить и сделать отдельный дашборд для мониторинга быстродействия и ответов upstream. Но это уже будет отдельная штука. А здесь у меня представлена общая информация для первичного анализа.