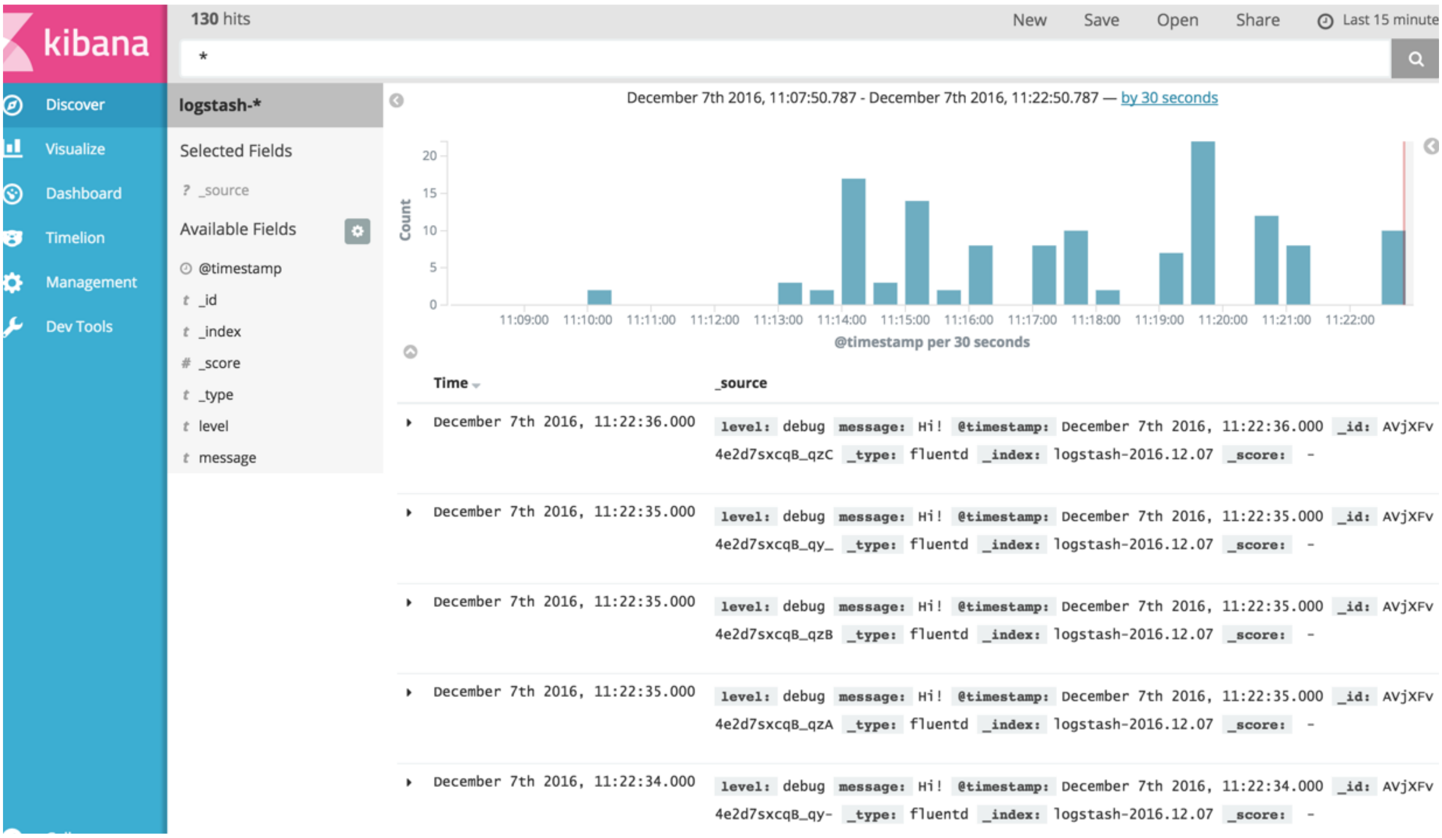
Docker Logging Efk Compose

This article explains how to collect Docker logs to EFK (Elasticsearch + Fluentd + Kibana) stack. The example uses Docker Compose for setting up multiple containers.



```
December 7th 2016, 11:22:36.000  level: debug message: Hi! @timestamp: December 7th 2016, 11:22:36.000 _id: AVjXFv
                                 4e2d7sxcqB_qzC _type: fluentd _index: logstash-2016.12.07 _score: -

December 7th 2016, 11:22:35.000  level: debug message: Hi! @timestamp: December 7th 2016, 11:22:35.000 _id: AVjXFv
                                 4e2d7sxcqB_qy_ _type: fluentd _index: logstash-2016.12.07 _score: -

December 7th 2016, 11:22:35.000  level: debug message: Hi! @timestamp: December 7th 2016, 11:22:35.000 _id: AVjXFv
                                 4e2d7sxcqB_qzB _type: fluentd _index: logstash-2016.12.07 _score: -

December 7th 2016, 11:22:35.000  level: debug message: Hi! @timestamp: December 7th 2016, 11:22:35.000 _id: AVjXFv
                                 4e2d7sxcqB_qzA _type: fluentd _index: logstash-2016.12.07 _score: -

December 7th 2016, 11:22:34.000  level: debug message: Hi! @timestamp: December 7th 2016, 11:22:34.000 _id: AVjXFv
                                 4e2d7sxcqB_qy- _type: fluentd _index: logstash-2016.12.07 _score: -
```

Elasticsearch is an open source search engine known for its ease of use. Kibana is an open source Web UI that makes Elasticsearch user friendly for marketers, engineers and data scientists alike.

By combining these three tools EFK (Elasticsearch + Fluentd + Kibana) we get a scalable, flexible, easy to use log collection and analytics pipeline. In this article, we will set up 4 containers, each includes:

- Apache HTTP Server
- Fluentd
- Elasticsearch
- Kibana

All of `httpd`'s logs will be ingested into Elasticsearch + Kibana, via Fluentd.

Prerequisites: Docker

Please download and install Docker / Docker Compose. Well, that's it :)

- Docker Installation

Step 0: prepare docker-compose.yml

First, please prepare `docker-compose.yml` for Docker Compose. Docker Compose is a tool for defining and running multi-container Docker applications.

With the YAML file below, you can create and start all the services (in this case, Apache, Fluentd, Elasticsearch, Kibana) by one command.

```
version: '2'
services:
  web:
    image: httpd
    ports:
      - "80:80"
    links:
      - fluentd
    logging:
      driver: "fluentd"
      options:
        fluentd-address: localhost:24224
        tag: httpd.access

  fluentd:
    build: ./fluentd
    volumes:
      - ./fluentd/conf:/fluentd/etc
    links:
      - "elasticsearch"
    ports:
      - "24224:24224"
      - "24224:24224/udp"

  elasticsearch:
    image: elasticsearch
    expose:
      - 9200
    ports:
      - "9200:9200"
  kibana:
    image: kibana
    links:
      - "elasticsearch"
    ports:
      - "5601:5601"
```

`logging` section (check Docker Compose documentation) of `web` container specifies Docker Fluentd Logging Driver as a default container logging driver. All of the logs from `web` container will be automatically forwarded to host:port specified by `fluentd-address`.

Step 1: Prepare Fluentd image with your Config + Plugin

Then, please prepare `fluentd/Dockerfile` with the following content, to use Fluentd's official Docker image and additionally install Elasticsearch plugin.

```
# fluentd/Dockerfile
FROM fluent/fluentd:v0.12-debian
RUN ["gem", "install", "fluent-plugin-elasticsearch", "--no-rdoc", "--no-ri", "-
```

Then, please prepare Fluentd's configuration file `fluentd/conf/fluent.conf`. in_forward plugin is used for receive logs from Docker logging driver, and out_elasticsearch is for forwarding logs to Elasticsearch.

```
# fluentd/conf/fluent.conf
<source>
  @type forward
  port 24224
  bind 0.0.0.0
</source>
<match *.**>
  @type copy
  <store>
    @type elasticsearch
    host elasticsearch
    port 9200
    logstash_format true
    logstash_prefix fluentd
    logstash_dateformat %Y%m%d
    include_tag_key true
    type_name access_log
    tag_key @log_name
    flush_interval 1s
  </store>
  <store>
    @type stdout
  </store>
</match>
```

Step 2: Start Containers

Let's start all of the containers, with just one command.

```
$ docker-compose up
```

You can check to see if 4 containers are running by `docker ps` command.

```
$ docker ps
CONTAINER ID        IMAGE                          COMMAND                  CREATED
2d28323d77a3        httpd                          "httpd-foreground"       About ar
a1b15a7210f6        dockercomposeefk_fluentd       "/bin/sh -c 'exec ..."   About ar
01e43b191cc1        kibana                         "/docker-entrypoin..."   About ar
b7b439415898        elasticsearch                  "/docker-entrypoin..."   About ar
```

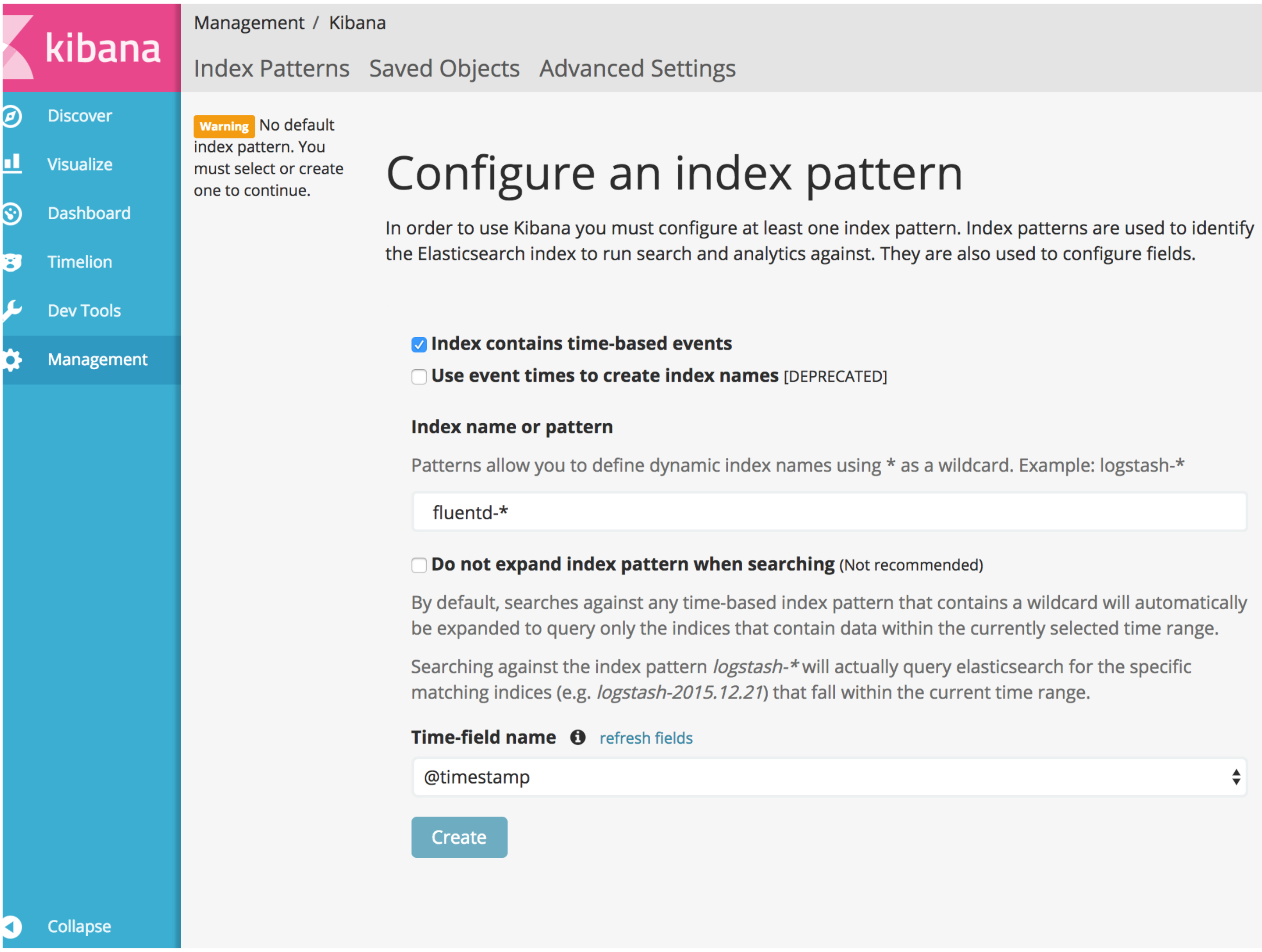
Step 3: Generate httpd Access Logs

Let's access to `httpd` to generate some access logs. `curl` command is always your friend.

```
$ repeat 10 curl http://localhost:80/
<html><body><h1>It works!</h1></body></html>
<html><body><h1>It works!</h1></body></html>
<html><body><h1>It works!</h1></body></html>
<html><body><h1>It works!</h1></body></html>
<html><body><h1>It works!</h1></body></html>
<html><body><h1>It works!</h1></body></html>
<html><body><h1>It works!</h1></body></html>
<html><body><h1>It works!</h1></body></html>
<html><body><h1>It works!</h1></body></html>
<html><body><h1>It works!</h1></body></html>
```

Step 4: Confirm Logs from Kibana

Please go to `http://localhost:5601/` with your browser. Then, you need to set up the index name pattern for Kibana. Please specify `fluentd-*` to `Index name or pattern` and press `Create` button.



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

Configure an index pattern

In order to use Kibana you must configure at least one index pattern. Index patterns are used to identify the Elasticsearch index to run search and analytics against. They are also used to configure fields.

☑ **Index contains time-based events**

□ **Use event times to create index names** [DEPRECATED]

Index name or pattern

Patterns allow you to define dynamic index names using * as a wildcard. Example: logstash-*

fluentd-*

□ **Do not expand index pattern when searching** (Not recommended)

By default, searches against any time-based index pattern that contains a wildcard will automatically be expanded to query only the indices that contain data within the currently selected time range.

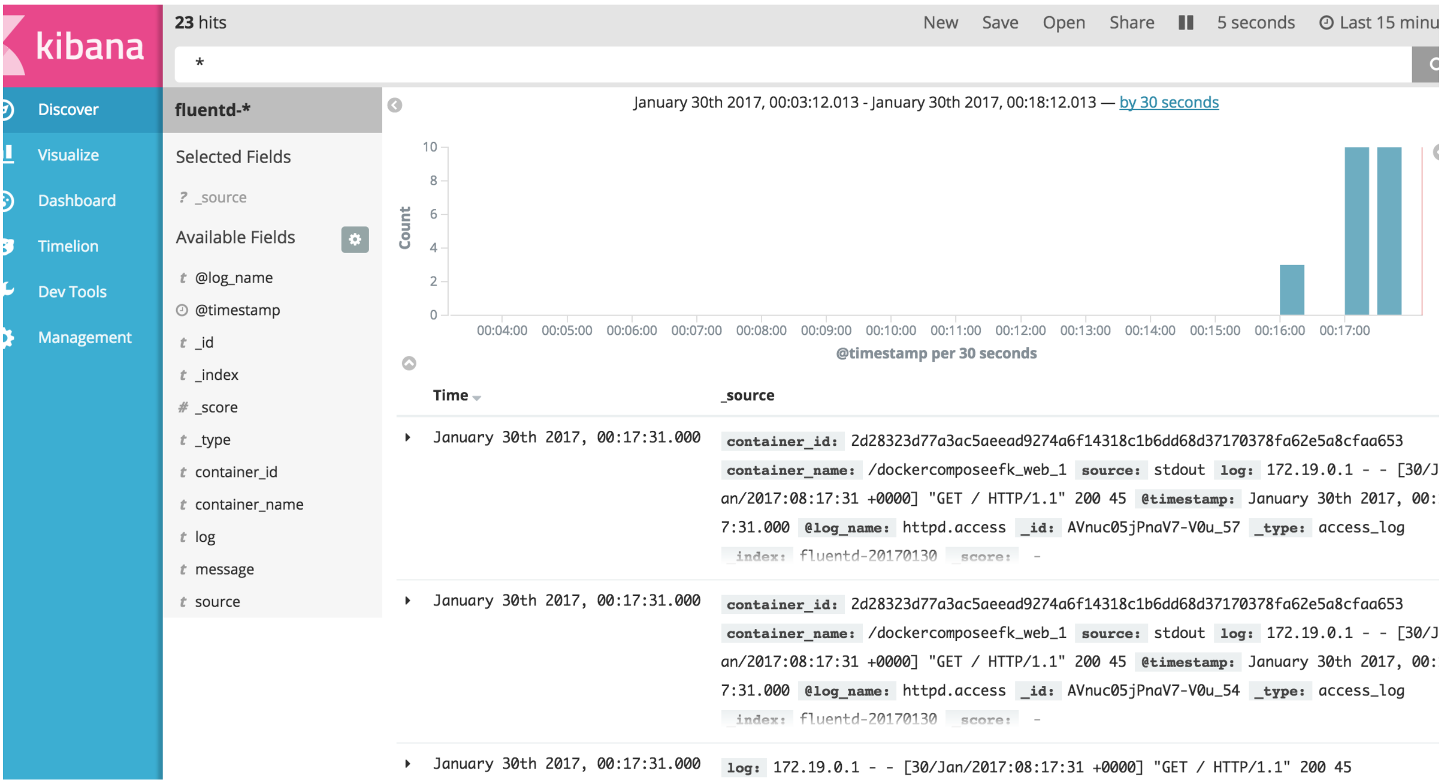
Searching against the index pattern *logstash-** will actually query elasticsearch for the specific matching indices (e.g. *logstash-2015.12.21*) that fall within the current time range.

Time-field name refresh fields

@timestamp

Create

Then, go to `Discover` tab to seek for the logs. As you can see, logs are properly collected into Elasticsearch + Kibana, via Fluentd.



23 hits

*

fluentd-*

January 30th 2017, 00:03:12.013 - January 30th 2017, 00:18:12.013 — by 30 seconds

```
Time                               _source
January 30th 2017, 00:17:31.000    container_id: 2d28323d77a3ac5aeead9274a6f14318c1b6dd68d37170378fa62e5a8cfaa653
                                   container_name: /dockercomposeefk_web_1 source: stdout log: 172.19.0.1 - - [30/J
                                   an/2017:08:17:31 +0000] "GET / HTTP/1.1" 200 45 @timestamp: January 30th 2017, 00:
                                   7:31.000 @log_name: httpd.access _id: AVnuc05jPnaV7-V0u_57 _type: access_log
                                   _index: fluentd-20170130 _score: -
January 30th 2017, 00:17:31.000    container_id: 2d28323d77a3ac5aeead9274a6f14318c1b6dd68d37170378fa62e5a8cfaa653
                                   container_name: /dockercomposeefk_web_1 source: stdout log: 172.19.0.1 - - [30/J
                                   an/2017:08:17:31 +0000] "GET / HTTP/1.1" 200 45 @timestamp: January 30th 2017, 00:
                                   7:31.000 @log_name: httpd.access _id: AVnuc05jPnaV7-V0u_54 _type: access_log
                                   _index: fluentd-20170130 _score: -
January 30th 2017, 00:17:31.000    log: 172.19.0.1 - - [30/Jan/2017:08:17:31 +0000] "GET / HTTP/1.1" 200 45
```

```
container_id: 2d28323d77a3ac5aeead9274a6f14318c1b6dd68d37170378fa62e5a8cfaa653
container_name: /dockercomposeefk_web_1 source: stdout @timestamp: January 30th
2017, 00:17:31.000 @log_name: httpd.access _id: AVnuc05jPnaV7-V0u_56 _type: acce
ss_log _index: fluentd-20170130 _score: -
```

Conclusion

This article explains how to collect logs from Apache to EFK (Elasticsearch + Fluentd + Kibana). The example code is available in this repository.

- https://github.com/kzk/docker-compose-efk

Learn More

- Fluentd Architecture
- Fluentd Get Started
- Downloading Fluentd

If this article is incorrect or outdated, or omits critical information, please let us know. Fluentd is a open source project under Cloud Native Computing Foundation (CNCF). All components are available under the Apache 2 License.

Previous
Common Log Formats

Next
Docker Logging

Last modified 3yr ago

WAS THIS PAGE HELPFUL?