TÌM HIỂU VỀ ELK (ELASTICSEARCH, LOGSTASH, KIBANA)

Người nộp: Phạm Mạnh Hà

**I. Tổng quan.**

**1. Log analysis (Phân tích bản ghi, nhật ký)**.

- Log cung cấp cho chúng ta những thông tin về cách hệ thống của chúng ta hoạt động như thế nào. Tuy nhiên, nội dung và định dạng của các log khác nhau của các dịch vụ khác nhau, giữa các thành phần khác nhau của hệ thống. Ví dụ một máy chủ web ghi lại thông tin về tất cả các yêu cầu đến, phản hồi đi, thời gian trả lời,...

Một số trường hợp sử dụng phân tích log mà nó hữu dụng:

* Vấn đề gỡ lỗi (debugging).
* Phân tích hiệu suất. (performance analysis).
* Phân tích bảo mật (security analysis).
* Phân tích chẩn đoán (predictive analysis).
* IoT và đăng nhập.

**Vấn đề gỡ lỗi (debugging).**

Gỡ lỗi là một trong những lý do phổ biến nhất để cho phép đăng nhập trong ứng dụng của bạn. Việc sử dụng đơn giản và thường xuyên nhất cho 1 log gỡ lỗi là grep cho một thông báo lỗi cụ thể hoặc xảy ra sự kiện. Nếu quản trị viên hệ thống cho rằng chương trình bị sập do lỗi mạng, thì người đó sẽ cố gắng tìm một tin nhắn connection dropped hoặc tin nhắn tương tự trong log máy chủ để phân tích nguyên nhân gây ra sự cố. Khi lỗi hoặc sự cố được xác định, các giải pháp phân tích log giúp nắm bắt thông tin ứng dụng và ảnh chụp nhanh về thời gian cụ thể đó có thể dễ dàng được chuyển qua các nhóm phát triển để phân tích thêm.

**Phân tích hiệu suất (performance analysis)**.

Phân tích log giúp tối ưu hóa hoặc gỡ lỗi hiệu năng hệ thống và đưa ra các đầu vào thiết yếu xung quanh các nút thắt trong hệ thống. Hiểu hiệu năng của hệ thống thường là hiểu về việc sử dụng tài nguyên trong hệ thống. Log có thể giúp phân tích việc sử dụng tài nguyên cá nhân trong hệ thống, hành vi của nhiều luồng trong ứng dụng, các điều kiện bế tắc tiềm năng, v.v. Log cũng mang theo dấu thời gian thông tin, đó là điều cần thiết để phân tích cách hệ thống hoạt động theo thời gian. Chẳng hạn, log của 1 máy chủ web có thể giúp biết các dịch vụ riêng lẻ đang hoạt động như thế nào dựa trên thời gian phản hồi, mã phản hồi HTTP, v.v.

**Phân tích bảo mật (security analysis)**.

Log đóng một vai trò quan trọng trong việc quản lý bảo mật ứng dụng cho bất kỳ tổ chức nào. Chúng đặc biệt hữu ích để phát hiện các vi phạm bảo mật, lạm dụng ứng dụng, tấn công độc hại, v.v. Khi người dùng tương tác với hệ thống, nó sẽ tạo ra các log sự kiện, có thể giúp theo dõi hành vi của người dùng, xác định các hoạt động đáng ngờ và đưa ra cảnh báo hoặc sự cố bảo mật cho các vi phạm.

Quá trình phát hiện xâm nhập liên quan đến việc xây dựng lại phiên từ chính các bản ghi. Ví dụ: các sự kiện đăng nhập ssh trong hệ thống có thể được sử dụng để xác định bất kỳ vi phạm nào trên máy.

**Phân tích chẩn đoán (predictive analysis)**.

Phân tích dự đoán là một trong những xu hướng nóng của thời gian gần đây. Log và dữ liệu sự kiện có thể được sử dụng để phân tích dự đoán rất chính xác. Mô hình phân tích dự đoán giúp trong việc xác định khách hàng tiềm năng, lập kế hoạch tài nguyên, quản lý và tối ưu hóa hàng tồn kho, hiệu quả khối lượng công việc và lập kế hoạch tài nguyên hiệu quả. Nó cũng hướng dẫn chiến lược tiếp thị, nhắm mục tiêu phân khúc người dùng, chiến lược vị trí quảng cáo, v.v.

**IoT và đăng nhập**.

Khi nói đến các thiết bị IoT (thiết bị hoặc máy tương tác với nhau mà không có sự can thiệp của con người), điều quan trọng là hệ thống phải được theo dõi và xử lý để giảm thời gian dừng và giải quyết nhanh chóng mọi lỗi hoặc sự cố quan trọng. Vì các thiết bị này sẽ có thể hoạt động với ít sự can thiệp của con người và có thể tồn tại ở quy mô địa lý lớn, log dữ liệu sẽ đóng một vai trò quan trọng trong việc hiểu hành vi của hệ thống và giảm thời gian chết.

**Những thách thức trong phân tích log**.

Quá trình phân tích log hiện tại chủ yếu liên quan đến việc kiểm tra log tại nhiều máy chủ được viết bởi các thành phần và hệ thống khác nhau trên ứng dụng của bạn. Điều này có nhiều vấn đề khác nhau, khiến nó trở thành một công việc tốn thời gian và tẻ nhạt. Hãy xem xét một số tình huống phổ biến:

• Định dạng log không nhất quán.

• Log phi tập trung (Decentralized logs).

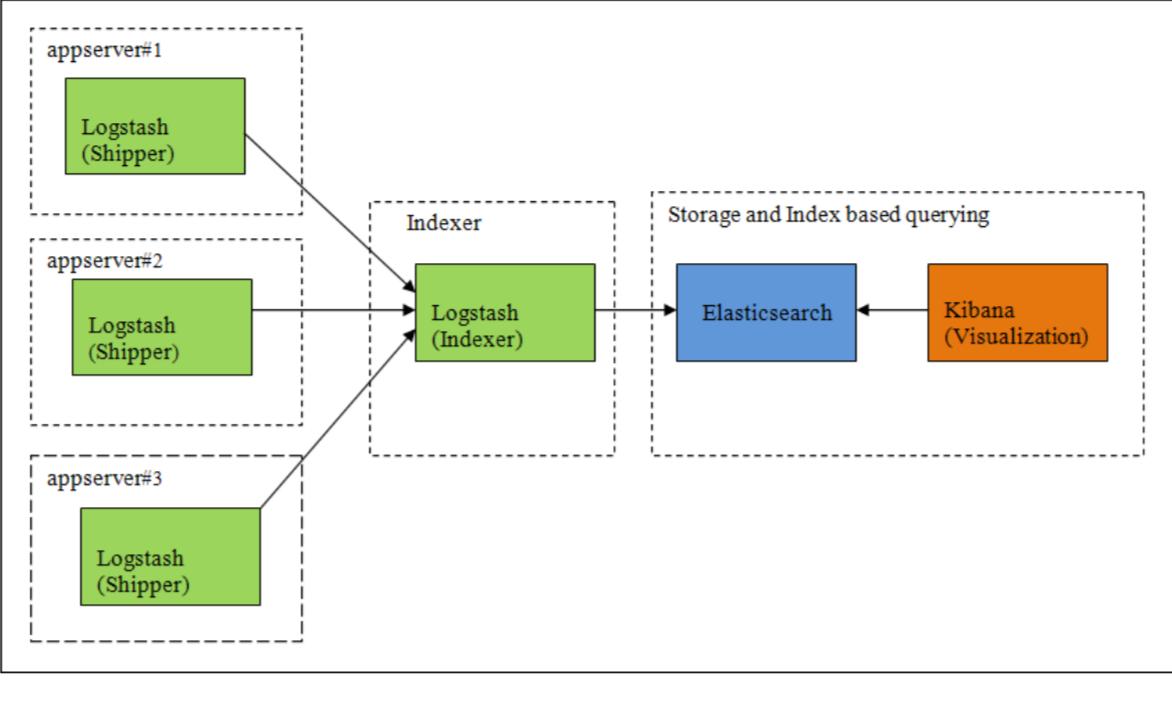
• Yêu cầu kiến thức chuyên môn.

Một điều thú vị để quan sát sẽ là cách các log khác nhau biểu thị các định dạng dấu thời gian khác nhau, các cách khác nhau để thể hiện INFO, ERROR, v.v. và thứ tự của các thành phần này với các log. Thật khó để tìm ra chỉ bằng cách xem log những gì hiện diện ở vị trí nào. Đây là điểm để các công cụ như Logstash trợ giúp.

**II. The ELK Stack**.

Nền tảng ELK là một giải pháp phân tích log hoàn chỉnh, được xây dựng trên sự kết hợp của ba công cụ nguồn mở - Elaticsearch, Logstash và Kibana. Nó cố gắng giải quyết tất cả các vấn đề và thách thức mà chúng ta đã thấy trong phần trước. ELK sử dụng ngăn xếp mã nguồn mở của Elaticsearch để tìm kiếm sâu và phân tích dữ liệu; Đăng nhập để quản lý ghi nhật ký tập trung, bao gồm vận chuyển và chuyển tiếp log từ nhiều máy chủ, làm đầy log và phân tích cú pháp; và cuối cùng, Kibana cho trực quan hóa dữ liệu mạnh mẽ và đẹp mắt. ELK stack hiện đang được duy trì và hỗ trợ tích cực bởi công ty có tên là Elastic (trước đây là Elaticsearch).

**ELK data pipeline**.



Trong một đường ống dữ liệu ELK Stack điển hình, các bản ghi từ nhiều máy chủ ứng dụng được chuyển qua người gửi Logstash đến một bộ chỉ mục Logstash tập trung. Trình lập chỉ mục Logstash sẽ xuất dữ liệu sang cụm Elaticsearch, được Kibana truy vấn để hiển thị trực quan hóa tuyệt vời và xây dựng bảng điều khiển trên log dữ liệu.

Hãy xem xét về từng hệ thống:

* Elasticsearch.
* Kibana.
* Logstash.

**1. Elasticsearch.**

Elaticsearch là một công cụ tìm kiếm nguồn mở phân tán dựa trên Apache Lucene và được phát hành theo giấy phép Apache 2.0 (có nghĩa là nó có thể được tải xuống, sử dụng và sửa đổi miễn phí). Nó cung cấp khả năng mở rộng theo chiều ngang, độ tin cậy và khả năng đa nhiệm cho tìm kiếm thời gian thực. Các tính năng tìm kiếm thông tin có sẵn thông qua JSON qua API RESTful. Các khả năng tìm kiếm được hỗ trợ bởi Apache Lucene Engine, không cho phép nó lập chỉ mục động dữ liệu mà không cần biết cấu trúc trước. Elaticsearch có thể đạt được các phản hồi tìm kiếm nhanh vì nó sử dụng lập chỉ mục để tìm kiếm trên các văn bản.

**2. Logstash**.

Logstash là một đường ống dữ liệu giúp thu thập, phân tích và phân tích một lượng lớn dữ liệu và sự kiện có cấu trúc và không cấu trúc được tạo ra trên các hệ thống khác nhau. Nó cung cấp các plugin để kết nối với nhiều loại nguồn và nền tảng đầu vào khác nhau và được thiết kế để xử lý hiệu quả các bản ghi, sự kiện và nguồn dữ liệu phi cấu trúc để phân phối thành nhiều loại đầu ra với việc sử dụng các plugin đầu ra của nó, cụ thể là tệp, đầu ra (như đầu ra trên bảng điều khiển đang chạy Logstash) hoặc Elaticsearch.

Nó có các tính năng sau:

* Xử lý dữ liệu tập trung: Logstash giúp xây dựng một đường ống dữ liệu có thể tập trung xử lý dữ liệu. Với việc sử dụng nhiều plugin cho đầu vào và đầu ra, nó có thể chuyển đổi rất nhiều nguồn đầu vào khác nhau thành một định dạng chung.
* Hỗ trợ cho các định dạng log tùy chỉnh: Log được viết bởi các ứng dụng khác nhau thường có các định dạng cụ thể dành riêng cho ứng dụng. Logstash giúp phân tích cú pháp và xử lý các định dạng tùy chỉnh trên quy mô lớn. Nó cung cấp hỗ trợ để viết các bộ lọc của riêng bạn cho mã thông báo và cũng cung cấp các bộ lọc sẵn sàng sử dụng.
* Phát triển plugin: Các plugin tùy chỉnh có thể được phát triển và xuất bản, và

có rất nhiều plugin được phát triển tùy chỉnh đã có sẵn.

**3. Kibana**.

Kibana là một nền tảng trực quan hóa dữ liệu được cấp phép Apache 2.0, giúp trực quan hóa mọi loại dữ liệu có cấu trúc và không cấu trúc được lưu trữ trong các chỉ mục của Elaticsearch. Kibana hoàn toàn được viết bằng HTML và JavaScript. Nó sử dụng các khả năng tìm kiếm và lập chỉ mục mạnh mẽ của Elaticsearch được thể hiện thông qua API RESTful của nó để hiển thị đồ họa mạnh mẽ cho người dùng cuối. Từ trí tuệ kinh doanh cơ bản đến gỡ lỗi thời gian thực, Kibana đóng vai trò của mình thông qua việc hiển thị dữ liệu thông qua biểu đồ đẹp, địa mạo, biểu đồ hình tròn, biểu đồ, bảng, v.v.

Kibana giúp dễ dàng hiểu được khối lượng dữ liệu lớn. Giao diện dựa trên trình duyệt đơn giản của nó cho phép bạn nhanh chóng tạo và chia sẻ bảng điều khiển động hiển thị các thay đổi đối với các truy vấn Elaticsearch trong thời gian thực.

Một số chức năng của Kibana:

• Nó cung cấp các phân tích linh hoạt và một nền tảng trực quan hóa cho doanh nghiệp

Sự thông minh.

• Nó cung cấp phân tích thời gian thực, tóm tắt, biểu đồ và gỡ lỗi

khả năng.

• Nó cung cấp một giao diện trực quan và thân thiện với người dùng, rất cao

có thể tùy chỉnh thông qua một số tính năng kéo và thả và sắp xếp

Khi cần thiết.

• Nó cho phép lưu bảng điều khiển và quản lý nhiều hơn một bảng điều khiển.

Bảng điều khiển có thể dễ dàng chia sẻ và nhúng trong các hệ thống khác nhau.

• Nó cho phép chia sẻ ảnh chụp nhanh của log mà bạn đã tìm kiếm thông qua,

và cô lập nhiều giao dịch có vấn đề.

**4. Cài đặt ELK Stack**.

Cần có thời gian chạy Java để chạy ELK Stack. Phiên bản mới nhất của Java được khuyến nghị cho việc cài đặt. Bạn có thể sử dụng phân phối chính thức của Oracle hoặc phân phối nguồn mở, chẳng hạn như OpenJDK.

$ java -version

openjdk version "1.8.0\_212"

OpenJDK Runtime Environment (build 1.8.0\_212-8u212-b03-0ubuntu1.18.04.1-b03)

OpenJDK 64-Bit Server VM (build 25.212-b03, mixed mode)

Nếu bạn đã xác minh cài đặt Java trong hệ thống của mình, chúng ta có thể tiến hành cài đặt ELK.

**4.1 Cài đặt Elasticsearch.**

Để bắt đầu, hãy chạy lệnh sau để nhập khóa GPG công khai của Elaticsearch vào APT:

$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -

Tiếp theo, thêm danh sách nguồn Elastic vào thư mục sources.list.d.

$ echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list

Tiếp theo, cập nhật danh sách gói của bạn để APT sẽ đọc nguồn Elastic mới:

$ sudo apt-get update

Sau đó tiến hành cài đặt Elasticsearch.

$ sudo apt-get install elasticsearch

Sau khi hoàn tất việc cài Elasticsearch, chỉnh sửa file cấu hình của Elasticsearch /etc/elasticsearch/elasticsearch.yml

Elaticsearch lắng nghe lưu lượng truy cập trên cổng 9200. Bạn sẽ muốn hạn chế quyền truy cập bên ngoài vào đối tượng Elaticsearch của mình để ngăn người ngoài đọc dữ liệu của bạn hoặc tắt Elaticsearch cluster của bạn thông qua API REST. Tìm dòng chỉ định network.host, bỏ ghi chú và thay thế giá trị của nó bằng localhost.

network.host: localhost

network.public\_host: địa chỉ IP

Phần port:

http.port: 9200

transport.tcp.port: 9300

Phần discovery:

discovery.type: single-node

Lưu và thoát.

Khởi động Elasticsearch. $ systemctl start elasticsearch

Kiểm tra Elasticsearch đã chạy hay chưa bằng cách gửi 1 HTTP request.

$ curl -X GET "localhost:9200"

(cần ảnh)

**4.2 Cài đặt Kibana**.

bạn chỉ nên cài đặt Kibana sau khi cài đặt Elaticsearch. Cài đặt theo thứ tự này đảm bảo rằng các thành phần mỗi sản phẩm phụ thuộc vào đúng vị trí.

Vì bạn đã thêm nguồn gói Elasticsearch ở bước trước, bạn chỉ có thể cài đặt các thành phần còn lại của Elastic Stack bằng apt:

$ sudo apt-get install kibana

Sau đó kích hoạt và khởi động Kibana

$ sudo systemctl enable kibana

$ sudo systemctl start kibana

Vì Kibana được cấu hình để chỉ nghe trên localhost, chúng ta phải thiết lập reserve proxy để cho phép truy cập bên ngoài vào nó. Sử dụng Nginx.

Đầu tiên, sử dụng lệnh openssl để tạo người dùng Kibana quản trị mà bạn sẽ sử dụng để truy cập vào giao diện web Kibana. Ví dụ đặt tên tài khoản là kibanaadmin.

Lệnh sau sẽ tạo người dùng và mật khẩu Kibana quản trị và lưu trữ chúng trong tệp htpasswd.users. Bạn sẽ định cấu hình Nginx để yêu cầu tên người dùng và mật khẩu này và đọc tệp này trong giây lát:

echo "kibanaadmin:`openssl passwd abc123`" | sudo tee -a /etc/nginx/htpasswd.users

Tiếp theo, tạo một tệp khối máy chủ Nginx. Ví dụ, đề cập đến tệp này là example123.com, mặc dù bạn có thể thấy hữu ích khi đặt cho bạn một tên mô tả hơn. Ví dụ: nếu có bản ghi FQDN và DNS được thiết lập cho máy chủ này, bạn có thể đặt tên tệp này theo tên FQDN của mình:

$ vi /etc/nginx/sites-available/example123.com

Thêm đoạn code sau vào file /etc/nginx/sites-available/example123.com, đảm bảo cập nhật example123.com để khớp với FQDN hoặc địa chỉ IP công cộng của máy chủ của bạn. Mã này định cấu hình Nginx để hướng lưu lượng HTTP của máy chủ của bạn đến ứng dụng Kibana, đang lắng nghe trên localhost: 5601. Ngoài ra, nó cấu hình Nginx để đọc tệp htpasswd.users và yêu cầu xác thực cơ bản.

server {

listen 80;

server\_name example.com;

auth\_basic "Restricted Access";

auth\_basic\_user\_file /etc/nginx/htpasswd.users;

location / {

proxy\_pass http://localhost:5601;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

Lưu và thoát.

Tiếp theo, kích hoạt cấu hình mới bằng cách tạo một liên kết tượng trưng đến thư mục sites-enabled. Nếu bạn đã tạo một tệp khối máy chủ có cùng tên trong điều kiện tiên quyết Nginx, bạn không cần phải chạy lệnh này:

$ sudo ln -s /etc/nginx/sites-available/example.com /etc/nginx/sites-enabled/example.com

Kiểm tra lỗi ký tự của file cấu hình bằng lệnh: $ sudo nginx -t

Không có lỗi thì hiển thị thông báo syntax is ok. Nếu chưa thì kiểm tra lại file cấu hình.

Khởi động lại Nginx: $ sudo systemctl restart nginx

Kibana hiện có thể truy cập thông qua FQDN hoặc địa chỉ IP của server Elastic Stack. Bạn có thể kiểm tra trang trạng thái của máy chủ Kibana bằng cách điều hướng đến địa chỉ sau và nhập thông tin đăng nhập của bạn khi được nhắc:

[**http://your\_server\_ip/status**](http://your_server_ip/status)

(cần ảnh)

**4.3 Cài đặt Logstash**.

Cài đặt Logstash bằng lệnh: $ sudo apt-get install logstash

Sau khi cài đặt Logstash, có thể chuyển sang cấu hình nó. Các tệp cấu hình của Logstash được viết theo định dạng JSON và nằm trong thư mục /etc/logstash/conf.d. Khi muốn cấu hình nó, sẽ hữu ích khi nghĩ rằng Logstash là một đường dẫn lấy dữ liệu ở một đầu, xử lý nó theo cách này hay cách khác và gửi nó đến đích của nó (trong trường hợp này, đích đến là Elaticsearch). Một đường dẫn Logstash có hai yếu tố bắt buộc, đầu vào và đầu ra, và một yếu tố tùy chọn, bộ lọc. Các plugin đầu vào tiêu thụ dữ liệu từ một nguồn, các plugin bộ lọc xử lý dữ liệu và các plugin đầu ra ghi dữ liệu đến đích.

(cần ảnh)

Tạo 1 file cấu hình là 02-beats-input.conf để cấu hình Filebeat input:

$ sudo vi /etc/logstash/conf.d/02-beats-input.conf

Chèn đoạn cấu hình input dưới đây vào file cấu hình, để xác định đầu vào sẽ nghe trên cổng TCP 5044

input {

beats {

port => 5044

}

}

Input plugins: được sử dụng để cấu hình một tập hợp các sự kiện sẽ được cung cấp cho Logstash.

Lưu cấu hình và thoát. Tiếp theo tạo file cấu hình 10-syslog-filter.conf để thêm bộ lọc (filter) vào system logs.

$ sudo vi /etc/logstash/conf.d/10-syslog-filter.conf

Chèn đoạn cấu hình bộ lọc sau:

filter {

if [fileset][module] == "system" {

if [fileset][name] == "auth" {

grok {

match => { "message" => ["%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} sshd(?:\[%{POSINT:[system][auth][pid]}\])?: %{DATA:[system][auth][ssh][event]} %{DATA:[system][auth][ssh][method]} for (invalid user )?%{DATA:[system][auth][user]} from %{IPORHOST:[system][auth][ssh][ip]} port %{NUMBER:[system][auth][ssh][port]} ssh2(: %{GREEDYDATA:[system][auth][ssh][signature]})?",

"%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} sshd(?:\[%{POSINT:[system][auth][pid]}\])?: %{DATA:[system][auth][ssh][event]} user %{DATA:[system][auth][user]} from %{IPORHOST:[system][auth][ssh][ip]}",

"%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} sshd(?:\[%{POSINT:[system][auth][pid]}\])?: Did not receive identification string from %{IPORHOST:[system][auth][ssh][dropped\_ip]}",

"%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} sudo(?:\[%{POSINT:[system][auth][pid]}\])?: \s\*%{DATA:[system][auth][user]} :( %{DATA:[system][auth][sudo][error]} ;)? TTY=%{DATA:[system][auth][sudo][tty]} ; PWD=%{DATA:[system][auth][sudo][pwd]} ; USER=%{DATA:[system][auth][sudo][user]} ; COMMAND=%{GREEDYDATA:[system][auth][sudo][command]}",

"%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} groupadd(?:\[%{POSINT:[system][auth][pid]}\])?: new group: name=%{DATA:system.auth.groupadd.name}, GID=%{NUMBER:system.auth.groupadd.gid}",

"%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} useradd(?:\[%{POSINT:[system][auth][pid]}\])?: new user: name=%{DATA:[system][auth][user][add][name]}, UID=%{NUMBER:[system][auth][user][add][uid]}, GID=%{NUMBER:[system][auth][user][add][gid]}, home=%{DATA:[system][auth][user][add][home]}, shell=%{DATA:[system][auth][user][add][shell]}$",

"%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} %{DATA:[system][auth][program]}(?:\[%{POSINT:[system][auth][pid]}\])?: %{GREEDYMULTILINE:[system][auth][message]}"] }

pattern\_definitions => {

"GREEDYMULTILINE"=> "(.|\n)\*"

}

remove\_field => "message"

}

date {

match => [ "[system][auth][timestamp]", "MMM d HH:mm:ss", "MMM dd HH:mm:ss" ]

}

geoip {

source => "[system][auth][ssh][ip]"

target => "[system][auth][ssh][geoip]"

}

}

else if [fileset][name] == "syslog" {

grok {

match => { "message" => ["%{SYSLOGTIMESTAMP:[system][syslog][timestamp]} %{SYSLOGHOST:[system][syslog][hostname]} %{DATA:[system][syslog][program]}(?:\[%{POSINT:[system][syslog][pid]}\])?: %{GREEDYMULTILINE:[system][syslog][message]}"] }

pattern\_definitions => { "GREEDYMULTILINE" => "(.|\n)\*" }

remove\_field => "message"

}

date {

match => [ "[system][syslog][timestamp]", "MMM d HH:mm:ss", "MMM dd HH:mm:ss" ]

}

}

}

}

Chú thích: Các plugin trong đoạn cấu hình trên có ý nghĩa gì?

date: Điều này được sử dụng để phân tích các trường ngày từ các sự kiện đến và sử dụng như các trường dấu thời gian Logstash.

grok: Đây là bộ lọc mạnh nhất để phân tích dữ liệu phi cấu trúc từ log hoặc sự kiện sang định dạng có cấu trúc.

Geoip: Điều này thêm thông tin địa lý dựa trên địa chỉ IP được truy xuất từ Maxmind database.

Lưu cấu hình và thoát. Tiếp theo tạo 1 file cấu hình 30-elasticsearch-output.conf

$ sudo vi /etc/logstash/conf.d/30-elasticsearch-output.conf

Chèn đoạn cấu hình output dưới đây vào file. Về cơ bản, đầu ra này cấu hình Logstash để lưu trữ dữ liệu Beats trong Elaticsearch, đang chạy tại localhost: 9200, trong một chỉ mục được đặt tên theo Beat được sử dụng.

output {

elasticsearch {

hosts => ["localhost:9200"]

manage\_template => false

index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"

}

}

Lưu cấu hình và thoát. Nếu bạn muốn thêm bộ lọc cho các ứng dụng khác sử dụng đầu vào Filebeat, hãy nhớ đặt tên cho các tệp để chúng được sắp xếp giữa đầu vào và cấu hình đầu ra, nghĩa là tên tệp phải bắt đầu bằng số có hai chữ số từ 02 đến 30.

Kiểm tra cấu hình Logstash đã được hay chưa, chạy dòng lệnh sau:

sudo -u logstash /usr/share/logstash/bin/logstash --path.settings /etc/logstash -t

Nếu không có lỗi cú pháp gì thì sẽ có thông báo Configuration OK, nó sẽ hiện ra 1 khoảng thời gian kiểm tra.

Start và enable Logstash:

$ sudo systemctl start logstash

$ sudo systemctl enable logstash

**4.4 Cài đặt Filebeat**.

ELK Stack sử dụng 1 vài phần mềm gửi dữ liệu gọi là Beats để thu thập dữ liệu từ nhiều nguồn khác nhau và gửi về Logstash hoặc Elasticsearch. Có các Beats sẵn có từ Elastic, nhưng ở bài viết này hướng dẫn cài đặt và sử dụng Filebeat.

Cài đặt Filebeat:$ sudo apt install filebeat

Tiếp theo cấu hình Filebeat kết nối đến Logstash. Cấu hình file /etc/filebeat/filebeat.yml như sau:

$ sudo vi /etc/filebeat/filebeat.yml

Filebeat hỗ trợ nhiều kết quả output, nhưng bạn sẽ chỉ gửi các sự kiện trực tiếp đến Elaticsearch hoặc Logstash để xử lý bổ sung. Trong hướng dẫn này sử dụng Logstash để thực hiện xử lý bổ sung trên dữ liệu do Filebeat thu thập. Filebeat sẽ không cần gửi bất kỳ dữ liệu nào trực tiếp đến Elaticsearch, vì vậy hãy vô hiệu hóa đầu ra đó. Để làm như vậy, hãy tìm phần output.elaticsearch và nhận xét các dòng sau bằng cách đặt trước chúng bằng dấu #:

......

#output.elasticsearch:

# Array of hosts to connect to.

#hosts: ["localhost:9200"]

.......

Sau đó, cấu hình phần output.logstash. Bỏ ghi chú các dòng output.logstash: và hosts: ["localhost: 5044"] bằng cách xóa #. Điều này sẽ cấu hình Filebeat để kết nối với Logstash trên máy chủ Elastic Stack tại cổng 5044, cổng đã chỉ định đầu vào Logstash trước đó:

......

output.logstash:

# The Logstash hosts

hosts: ["localhost:5044"]

......

Lưu cấu hình và thoát. Chức năng của Filebeat có thể được mở rộng với các mô-đun Filebeat. Trong hướng dẫn này, chúng tôi sẽ sử dụng mô-đun hệ thống, thu thập và phân tích các bản ghi được tạo bởi dịch vụ ghi log hệ thống của các bản phân phối Linux phổ biến.

$ sudo filebeat modules enable system

Tiếp theo, tải mẫu chỉ mục vào Elaticsearch. Một chỉ mục Elaticsearch là một tập hợp các tài liệu có các đặc điểm tương tự. Các chỉ mục được xác định bằng một tên, được sử dụng để chỉ chỉ mục khi thực hiện các hoạt động khác nhau trong đó. Mẫu chỉ mục sẽ được tự động áp dụng khi tạo chỉ mục mới.

Để tải chỉ mục, sử dụng lệnh sau:

$ sudo filebeat setup --template -E output.logstash.enabled=false -E 'output.elasticsearch.hosts=["localhost:9200"]'

Filebeat được đóng gói với bảng điều khiển Kibana mẫu cho phép bạn hình dung dữ liệu Filebeat trong Kibana. Trước khi bạn có thể sử dụng bảng điều khiển, bạn cần tạo mẫu chỉ mục và tải bảng điều khiển vào Kibana.

Khi bảng điều khiển tải, Filebeat kết nối với Elaticsearch để kiểm tra thông tin phiên bản. Để tải bảng điều khiển khi Logstash được bật, bạn cần tắt đầu ra Logstash và bật đầu ra Elaticsearch:

$ sudo filebeat setup -e -E output.logstash.enabled=false -E output.elasticsearch.hosts=['localhost:9200'] -E setup.kibana.host=localhost:5601

Giờ có thể khởi động filebeat:

$ sudo systemctl start filebeat

$ sudo systemctl enable filebeat

Nếu bạn đã thiết lập Elastic Stack chính xác, Filebeat sẽ bắt đầu chuyển nhật ký log hệ thống và syslog của bạn tới Logstash, sau đó sẽ tải dữ liệu đó vào Elaticsearch.

Để xác minh rằng Elaticsearch thực sự đang nhận dữ liệu này, hãy truy vấn chỉ mục Filebeat bằng lệnh này:

$ curl -XGET 'http://localhost:9200/filebeat-\*/\_search?pretty'

Sau khi chạy lệnh trên, sẽ hiện ra kết quả tương tự như sau:

...

{

"took" : 32,

"timed\_out" : false,

"\_shards" : {

"total" : 3,

"successful" : 3,

"skipped" : 0,

"failed" : 0

},

"hits" : {

"total" : 1641,

"max\_score" : 1.0,

"hits" : [

{

"\_index" : "filebeat-6.4.2-2018.10.10",

"\_type" : "doc",

"\_id" : "H\_bZ62UBB4D0uxFRu\_h3",

"\_score" : 1.0,

"\_source" : {

"@version" : "1",

"message" : "Oct 10 06:22:36 elk systemd[1]: Reached target Local File Systems (Pre).",

"@timestamp" : "2018-10-10T08:43:56.969Z",

"host" : {

"name" : "elk"

},

"source" : "/var/log/syslog",

"input" : {

"type" : "log"

},

"tags" : [

"beats\_input\_codec\_plain\_applied"

],

"offset" : 296,

"prospector" : {

"type" : "log"

},

"beat" : {

"version" : "6.4.2",

"hostname" : "elk",

"name" : "elk"

}

}

},

...

**5. Thu thập, phân tích và chuyển đổi dữ liệu với Logstash.**

**- Logstash plugins.**

Logstash có nhiều plugin để giúp tích hợp nó với nhiều nguồn đầu vào và đầu ra.

**5.1 Kiểu dữ liệu cho plugin properties**.

Một plugin Logstash yêu cầu các cài đặt hoặc thuộc tính nhất định được đặt. Các thuộc tính này có các giá trị nhất định thuộc về một trong các loại dữ liệu quan trọng sau.

**6. Vai trò của Elasticsearch trong ELK**.

Elasticsearch là 1 công cụ tìm kiếm cho phép tìm kiếm nhanh rộng trong 1 môi trường phân tán. Như đã nói ở mục II, một số kiến trúc phân tán lớn như GitHub, StackOverflow và Wikipedia sử dụng tìm kiếm toàn văn bản, tìm kiếm có cấu trúc, khả năng phân tích nhanh để đưa ra tìm kiếm phù hợp.

**6.1 Khái niệm cơ bản về Elasticsearch**.

**+) Mục lục (index)**

Mục lục trong Elaticsearch là một tập hợp các tài liệu có chung một số đặc điểm chung.

Mỗi chỉ mục chứa nhiều loại, lần lượt chứa nhiều tài liệu và mỗi tài liệu chứa nhiều trường. Một chỉ mục bao gồm nhiều tài liệu JSON trong Elaticsearch. Có thể có bất kỳ số lượng chỉ mục nào trong một cụm trong Elaticsearch.

Trong ELK, khi JSON document của Logstash được gửi đến Elasticsearch, chúng được gửi đi như 1 chỉ mục mặc định dưới dạng "logstash-%{+YYYY.MM.dd}".Nó phân vùng các chỉ mục theo ngày để có thể dễ dàng tìm kiếm và xóa nếu cần. Mẫu này có thể được thay đổi trong cấu hình plugin đầu ra Logstash.

URL để tìm kiếm và truy vấn các chỉ mục có dạng như sau:

[http://localhost:9200/[index]/[type]/[operation](http://localhost:9200/%5Bindex%5D/%5Btype%5D/%5Boperation)]

**+) Document**

Một document trong Elaticsearch là một JSON document được lưu trữ trong một chỉ mục. Mỗi tài liệu có một loại và ID tương ứng, đại diện cho nó duy nhất. Ví dụ:

{

"\_index" : "packtpub",

"\_type" : "elk",

"\_id" : "1",

"\_version" : 1,

"found" : true,

"\_source":{

book\_name : "learning elk"

}

}

**+) Field (trường)**

Một trường là một đơn vị cơ bản trong một tài liệu. Như trong ví dụ trước, trường cơ bản là cặp giá trị khóa như sau: book\_name : "learning elk"

**+) Type (Kiểu)**

Loại được sử dụng để cung cấp một phân vùng hợp lý bên trong các chỉ mục. Về cơ bản nó đại diện cho một lớp các loại tài liệu tương tự. Một chỉ mục có thể có nhiều loại và chúng ta có thể định nghĩa chúng theo ngữ cảnh. Chẳng hạn, chỉ mục cho Facebook có thể có bài đăng là một trong những loại chỉ mục, bình luận loại khác.

**+) Mapping (Ánh xạ)**

Ánh xạ được sử dụng để ánh xạ từng trường của tài liệu với kiểu dữ liệu tương ứng của nó, chẳng hạn như chuỗi, số nguyên, float, double, date, v.v. Elaticsearch tạo tự động ánh xạ cho các trường trong quá trình tạo chỉ mục và những ánh xạ đó có thể dễ dàng được truy vấn hoặc sửa đổi dựa trên các loại nhu cầu cụ thể.

**+) Shard (Phân đoạn)**

Phân đoạn là thực thể vật lý thực tế nơi dữ liệu cho mỗi chỉ mục được lưu trữ. Mỗi chỉ mục có thể có một số phân đoạn chính và bản sao nơi lưu trữ dữ liệu. Các phân đoạn được phân phối giữa tất cả các nút trong cụm và có thể được di chuyển từ nút này sang nút khác trong trường hợp lỗi nút hoặc thêm các nút mới.

**+) Primary shard and replica shard (Phân đoạn chính và bản sao)**

Mỗi tài liệu trong một chỉ mục Elaticsearch được lưu trữ trên một phân đoạn chính và một số phân đoạn sao chép. Trong khi lập chỉ mục, tài liệu đầu tiên được lưu trữ trên phân đoạn chính và sau đó trên phân đoạn bản sao tương ứng. Theo mặc định, số lượng phân đoạn chính cho mỗi chỉ mục là năm và có thể được định cấu hình theo nhu cầu của chúng tôi.

Các phân đoạn sao chép thường sẽ nằm trên một nút khác với phân đoạn chính và trợ giúp trong trường hợp chuyển đổi dự phòng và cân bằng tải để phục vụ cho nhiều yêu cầu.

**+) Cluster (Cụm)**

Một cụm là một tập hợp các nút lưu trữ dữ liệu được lập chỉ mục. Elaticsearch cung cấp khả năng mở rộng theo chiều ngang với sự trợ giúp của dữ liệu được lưu trữ trong cụm. Mỗi cụm được đại diện bởi một tên cụm, mà các nút khác nhau tham gia. Tên cụm được đặt bởi một thuộc tính được gọi là cluster.name trong cấu hình elaticsearch.yml, mặc định là "elaticsearch":

cluster.name: elasticsearch

**+) Node (Nút)**

Một nút là một phiên bản chạy duy nhất của Elaticsearch, thuộc về một trong các cụm. Theo mặc định, mọi nút trong Elaticsearch đều tham gia cụm có tên là "elaticsearch". Mỗi nút có thể có cấu hình riêng được xác định trong elasticsearch.yml, chúng có thể có các cài đặt khác nhau về phân bổ tài nguyên và bộ nhớ.

Trong Elasticsearch, nút có 3 vai trò:

* **Data node:** Các nút dữ liệu lập chỉ mục tài liệu và thực hiện tìm kiếm trên các tài liệu được lập chỉ mục. Chúng tôi luôn khuyến nghị thêm nhiều nút dữ liệu để tăng hiệu suất hoặc chia tỷ lệ cụm. Một nút có thể được tạo thành một nút dữ liệu bằng cách đặt các thuộc tính này trong cấu hình elasticsearch.yml cho nút:

node.master = false

node.data=true

* **Master node:** master node Các nút chủ chịu trách nhiệm quản lý một cụm. Đối với các cụm lớn, nên có ba nút chính chuyên dụng (một chính và hai sao lưu), chỉ hoạt động như các nút chính và không lưu trữ các chỉ mục hoặc thực hiện tìm kiếm. Một nút có thể được cấu hình để trở thành một nút chủ chuyên dụng với cấu hình này trong elasticsearch.yml:

node.master =true

node.data=false

* **Routing node or load balancer node:** Các nút này không đóng vai trò là nút chính hoặc nút dữ liệu, mà chỉ thực hiện cân bằng tải hoặc định tuyến các yêu cầu tìm kiếm hoặc lập chỉ mục tài liệu cho các nút thích hợp. Điều này rất hữu ích cho các tìm kiếm khối lượng lớn hoặc hoạt động chỉ mục. Một nút có thể được cấu hình để trở thành một nút định tuyến với cấu hình này trong elasticsearch.yml:

node.master =false

node.data=false

**6.2 Khám phá API Elaticsearch.**

Trong ELK, mặc dù Logstash và Kibana hoạt động như một giao diện để nói chuyện với các chỉ số của Elaticsearch, nhưng vẫn cần phải hiểu cách Logstash và Kibana sử dụng API của Elaticsearch RESTful để thực hiện các hoạt động khác nhau, như tạo và quản lý các chỉ mục, lưu trữ và truy xuất tài liệu, và hình thành các loại truy vấn tìm kiếm khác nhau xung quanh các chỉ số. Nó cũng thường hữu ích để biết cách xóa các chỉ mục.

Elaticsearch cung cấp một API mở rộng để thực hiện các hoạt động khác nhau. Cú pháp chung của truy vấn cụm từ dòng lệnh như sau:

$curl -X<VERB>

'<PROTOCOL>://<HOST>:<PORT>/<PATH>/<OPERATION\_NAME>?<QUERY\_STRING>' -

d '<BODY>'

Khái niệm từng phần trong câu lệnh trên:

• VERB: Điều này có thể lấy các giá trị cho loại phương thức yêu cầu: GET, POST, PUT, DELETE, HEAD.

• PROTOCOL: Đây là http hoặc https.

• HOST: Đây là tên máy chủ của nút trong cụm. Đối với cài đặt cục bộ, đây có thể là 'localhost' hoặc '127.0.0.1'.

• PORT: Đây là cổng mà đối tượng Elaticsearch hiện đang chạy. Mặc định là 9200.

• PATH: Điều này tương ứng với tên của chỉ mục, loại và ID sẽ được truy vấn, ví dụ: / index / type / id.

• OPERATION\_NAME: Điều này tương ứng với tên của hoạt động sẽ được thực hiện, ví dụ: \_search, \_count, v.v.

• QUERY\_STRING: Đây là một tham số tùy chọn được chỉ định cho các tham số chuỗi truy vấn. Ví dụ :? pretty để in các tài liệu JSON.

• BODY: Điều này thực hiện một yêu cầu cho văn bản cơ thể.

Ví dụ: $ curl -X GET "[http://localhost:9200](http://localhosy:9200)"

**Trạng thái của cluster**.

• **Màu đỏ** biểu thị rằng một số hoặc tất cả các phân đoạn chính chưa sẵn sàng để phục vụ các yêu cầu.

**• Màu vàng** biểu thị rằng tất cả các phân đoạn chính được phân bổ nhưng một số hoặc tất cả các bản sao chưa được phân bổ. Thông thường, các cụm nút đơn sẽ có trạng thái màu vàng vì không có nút nào khác có sẵn để sao chép.

• **Màu xanh lá cây** biểu thị rằng tất cả các phân đoạn và bản sao của chúng được phân bổ tốt và cụm hoạt động đầy đủ.

**6.2 Tạo 1 chỉ mục.**

Trong ELK, việc tạo chỉ mục được tự động xử lý bằng cách cung cấp tên chỉ mục trong plugin đầu ra của Logstash elaticsearch. Tuy nhiên, hãy xem làm thế nào chúng ta có thể tạo một chỉ mục:

curl -XPUT 'localhost:9200/<index\_name>?pretty'

Ví dụ: để tạo một chỉ mục có tên packtpub, chúng ta có thể đưa ra lệnh sau:

curl –XPUT 'localhost:9200/packtpub/?pretty'

Có thể trực tiếp tạo một chỉ mục trong khi đặt tài liệu bên trong chỉ mục như sau:

curl –xPUT 'localhost:9200/packtpub/elk/1?pretty' –d '

{

book\_name : "learning elk"

}'

Kết quả trả lại có dạng:

{

"\_index" : "packtpub",

"\_type" : "elk",

"\_id" : "1",

"\_version" : 1,

"created" : true

}

**6.3 Lấy document**.

Bây giờ chúng tôi sẽ lấy tài liệu mà chúng tôi vừa lập chỉ mục:

curl -XGET 'localhost:9200/packtpub/elk/1?pretty'

Kết quả của truy vấn trước sẽ là:

{

"\_index" : "packtpub",

"\_type" : "elk",

"\_id" : "1",

"\_version" : 1,

"found" : true,

"\_source":{

book\_name : "learning elk"

}

}

**6.4. Xóa document**.

Để xóa tài liệu bên trong một chỉ mục, chúng tôi có thể đưa ra các điều sau đây

chỉ huy:

curl -XDELETE 'localhost:9200/packtpub?pretty'

**6.5 Xóa 1 index**.

curl -XDELETE 'localhost:9200/packtpub?pretty'

Kết quả trả về như sau:

{

"acknowledged" : true

}

Điều này chỉ ra rằng chỉ mục đã được xóa thành công.

**6.6. Truy vấn Elasticsearch DSL.**

Các truy vấn mà chúng tôi thấy cho đến bây giờ là các lệnh cơ bản được sử dụng để truy xuất dữ liệu, nhưng sức mạnh thực sự của truy vấn của Elaticsearch nằm trong Query Domain Specific Language dựa trên JSON cũng được gọi là Truy vấn DSL. Kibana sử dụng rộng rãi truy vấn DSL để có kết quả ở định dạng mong muốn cho bạn. Bạn gần như không bao giờ thực sự phải lo lắng về việc viết JSON truy vấn, vì Kibana sẽ tự động tạo và đặt kết quả ở định dạng đẹp.

Ví dụ: để chỉ nhận được ba kết quả trong số tất cả các kết quả phù hợp, chúng ta có thể

chỉ định nó như thế này:

curl -XPOST 'localhost:9200/logstash-\*/\_search' -d '

{

"query": { "match\_all": {} },

"size": 3

}'

Câu trả lời như sau, trong đó có ba tài liệu phù hợp với tìm kiếm:

{

"took" : 390,

"timed\_out" : false,

"\_shards" : {

"total" : 640,

"successful" : 640,

"failed" : 0

},

"hits" : {

"total" : 128,

"max\_score" : 1.0,

"hits" : [{

"\_index" : "logstash-2014.07.01",

"\_type" : "logs",

"\_id" : "AU2qge3cPoayDyQnreX0",

"\_score" : 1.0,

"\_source" : {

"message" : ["2014-07-

02,583.3526,585.44269,580.39264,582.33765,1056400,582.33765"],

"@version" : "1",

"@timestamp" : "2014-07-01T23:00:00.000Z",

"host" : "packtpub",

"path" : "/opt/logstash/input/GOOG.csv",

"date\_of\_record" : "2014-07-02",

"open" : 583.3526,

"high" : 585.44269,

"low" : 580.39264,

"close" : 582.33765,

"volume" : 1056400,

"adj\_close" : "582.33765"

}

}, {

"\_index" : "logstash-2014.07.09",

[ 97 ]

www.it-ebooks.infoWhy Do We Need Elasticsearch in ELK?

"\_type" : "logs",

"\_id" : "AU2qge3cPoayDyQnreXv",

"\_score" : 1.0,

"\_source" : {

"message" : ["2014-07-

10,565.91254,576.59265,565.01257,571.10254,1356700,571.10254"],

"@version" : "1",

"@timestamp" : "2014-07-09T23:00:00.000Z",

"host" : "packtpub",

"path" : "/opt/logstash/input/GOOG.csv",

"date\_of\_record" : "2014-07-10",

"open" : 565.91254,

"high" : 576.59265,

"low" : 565.01257,

"close" : 571.10254,

"volume" : 1356700,

"adj\_close" : "571.10254"

}

}, {

"\_index" : "logstash-2014.07.21",

"\_type" : "logs",

"\_id" : "AU2qgZixPoayDyQnreXn",

"\_score" : 1.0,

"\_source" : {

"message" : ["2014-07-

22,590.72266,599.65271,590.60266,594.74268,1699200,594.74268"],

"@version" : "1",

"@timestamp" : "2014-07-21T23:00:00.000Z",

"host" : "packtpub",

"path" : "/opt/logstash/input/GOOG.csv",

"date\_of\_record" : "2014-07-22",

"open" : 590.72266,

"high" : 599.65271,

"low" : 590.60266,

"close" : 594.74268,

"volume" : 1699200,

"adj\_close" : "594.74268"

}

}

]

}

}

**7. Tìm hiểu sâu về Kibana**.

Trong phần này, chúng ta sẽ xem Kibana đóng vai trò là đầu mối của ELK, nơi nó che giấu tất cả sự phức tạp của dữ liệu và trình bày các hình ảnh, biểu đồ và bảng điều khiển đẹp mắt được xây dựng trên dữ liệu, giúp hiểu rõ hơn về dữ liệu.

Kibana giúp dễ dàng tạo và chia sẻ bảng điều khiển bao gồm nhiều loại biểu đồ và đồ thị khác nhau. Trực quan hóa Kibana tự động hiển thị các thay đổi trong dữ liệu theo thời gian dựa trên các truy vấn Elaticsearch. Thật dễ dàng để cài đặt và thiết lập, đồng thời giúp chúng tôi nhanh chóng tìm hiểu và khám phá nhiều khía cạnh của dữ liệu.

Các tính năng của Kibana:

**Elasticsearch tổng hợp.**

Chủ yếu có hai loại tổng hợp - Bucketing và Metrics. Bucketing tạo ra một danh sách các nhóm, mỗi nhóm có một bộ tài liệu thuộc về nó, ví dụ: các điều khoản, phạm vi, biểu đồ, v.v. Số liệu tính toán số liệu tính toán cho một tập hợp các tài liệu, ví dụ: min, max, sum, trung bình, v.v. Các loại tính toán này chỉ có thể được thực hiện trên loại trường số.

**Scripted fields.**

Các trường script được sử dụng để thực hiện tính toán nhanh chóng trên dữ liệu được lập chỉ mục. Ví dụ: đối với một trường nhất định, bạn luôn muốn nhân với 100 trước khi hiển thị. Bạn có thể lưu nó dưới dạng một trường script. Các trường theo kịch bản, mặc dù, không thể được tìm kiếm.

Ví dụ về 1 script: doc['volume'].value \* 100 .

Kịch bản này sẽ luôn nhân giá trị với 100 trước khi nó hiển thị.

**Bảng điều khiển động**

Bảng điều khiển rất linh hoạt và năng động vì trực quan hóa cá nhân có thể được sắp xếp dễ dàng theo sự thuận tiện và dữ liệu có thể được làm mới tự động.

**Giao diện Kibana.**

Giao diện Kibana bao gồm bốn tab chính:

• **Discover:** Trang Khám phá cho phép tìm kiếm văn bản miễn phí, tìm kiếm dựa trên trường, tìm kiếm dựa trên phạm vi, v.v.

• **Visualize:** Trang Visualize cho phép xây dựng nhiều trực quan hóa, chẳng hạn như biểu đồ hình tròn, biểu đồ thanh, biểu đồ đường, v.v., có thể được lưu và sử dụng trong bảng điều khiển sau này.

• **Dashboard:** Bảng điều khiển thể hiện các bộ sưu tập nhiều trực quan hóa và tìm kiếm, có thể được sử dụng để dễ dàng áp dụng các bộ lọc dựa trên tương tác nhấp chuột và đưa ra kết luận dựa trên nhiều tập hợp dữ liệu.

• **Setting:** Cài đặt cho phép cấu hình các mẫu chỉ mục, các trường theo kịch bản, các kiểu dữ liệu của các trường, v.v.

**7.1 Chi tiết hơn về trang Discover.**

Discover được sử dụng để thực hiện tìm kiếm tương tác trên dữ liệu được lập chỉ mục của bạn. Nó cho phép bạn thực hiện tìm kiếm ad hoc dựa trên các trường, lọc dữ liệu và cho phép bạn xem các tài liệu được lập chỉ mục.

**+) Time filter**

Bạn có nhớ thời gian khi sếp của bạn yêu cầu tìm một số thống kê từ dữ liệu của bạn trong một thời gian cụ thể không? Bộ lọc thời gian là câu trả lời cho các loại tìm kiếm này. Bạn có thể lọc dữ liệu trên bất kỳ khoảng thời gian cụ thể nào được chọn từ lịch, được gọi là **Absolute** hoặc làm cho nó **Relative** dựa trên thời gian hiện tại. Ngoài ra còn có một số bộ lọc thời gian nhanh chóng có sẵn để sử dụng.

**7.1.1 Truy vấn và tìm kiếm dữ liệu.**

Kibana sử dụng cú pháp truy vấn Lucene để tìm kiếm giữa các chỉ mục được lưu trữ trong các mẫu chỉ mục.

1 vài ví dụ về tìm kiếm:

**+) Tìm kiếm nội dung (Freetext search)**

Tìm kiếm freetext nhằm mục đích lọc các tài liệu có chứa cụm từ tìm kiếm. Nó tìm kiếm trong tất cả các tài liệu cho tất cả các trường có chứa thuật ngữ tìm kiếm.

Hãy lấy ví dụ sau đây làm ví dụ:

Bạn muốn tìm kiếm tất cả các sách ELK từ một mẫu chỉ mục bao gồm các bộ sưu tập sách PacktPub. Bạn có thể viết 'ELK' trong hộp tìm kiếm và nó lọc tất cả các tài liệu có chứa thuật ngữ ELK.

Các tìm kiếm Boolean có thể được thực hiện theo các thuật ngữ sau:

**AND**

"Learning" AND "ELK"

Truy vấn trước sẽ tìm kiếm tất cả các tài liệu có chứa cả hai thuật ngữ:

"Learning" và "ELK"

**OR**

"Logstash" OR "ELK"

Truy vấn trước sẽ tìm kiếm tất cả các tài liệu có chứa 1 trong 2 thuật ngữ:

"Logstash" hoặc "ELK".

**NOT**

"Logstash" NOT "ELK"

Truy vấn trước sẽ tìm kiếm tất cả các tài liệu chứa thuật ngữ logstash chứ không phải thuật ngữ ELK:

**Groupings**

("Logstash" OR "ELK") AND "Kibana"

Truy vấn trước sẽ tìm kiếm các tài liệu có chứa "Kibana" và có thể chứa "ELK" hoặc "Logstash".

**Tìm kiếm ký tự đại diện**

Bạn cũng có thể thực hiện tìm kiếm ký tự đại diện bằng các thuật ngữ sau:

• plan \*: sẽ tìm kiếm tất cả các tài liệu có các điều khoản, chẳng hạn như plant, planting,v.v.

• plan? : sẽ tìm kiếm plant hoặc plans.

•? and \*: không thể được sử dụng làm ký tự đầu tiên trong tìm kiếm

**Tìm kiếm theo trường.**

Tìm kiếm trường nhằm tìm kiếm các giá trị cụ thể hoặc phạm vi giá trị cho các trường trong tài liệu được lập chỉ mục của bạn hiển thị ở phía bên trái của trang **Discover.**

Tìm kiếm trường có thể được thực hiện bằng tên trường và ký tự: theo sau là giá trị cho trường chúng tôi muốn lọc.

<field\_name>: <field\_value>

Ví dụ về tìm kiếm theo trường:

title : "Learning ELK"

title : "Learning ELK" AND category : "technology"

**Tìm kiếm rộng**

Tìm kiếm phạm vi được sử dụng để tìm kiếm một phạm vi giá trị cho một trường.

Ví dụ: để tìm kiếm một phạm vi ngày cụ thể:

date\_of\_record : [20140701 TO 20141231]

Để tìm kiếm giá trị trong trường volume:

volume : [ 100000 TO 200000]

Tìm kiếm phạm vi và trường có thể được kết hợp bằng cách sử dụng các toán tử boolean như thế này:

publish\_date : [20150701 TO 20151231] AND title : "Learning ELK"

Ký tự đặc biệt

Sau đây là danh sách các ký tự đặc biệt, nếu chúng ta muốn tìm kiếm, cần phải thoát bằng toán tử \: + - && || ! ( ) { } [ ] ^ " ~ \* ? : \

**Tìm kiếm trường bằng danh sách trường**.

Tìm kiếm trường cũng có thể được thực hiện bằng cách nhấp vào biểu tượng bộ lọc tích cực hoặc tiêu cực trên các giá trị nhất định trên trường.

**7.2 Visualize page.**

Trang **Visualize** giúp tạo trực quan hóa dưới dạng biểu đồ và biểu đồ. Những hình ảnh này có thể được lưu và xem riêng lẻ hoặc có thể được sử dụng trong nhiều bảng điều khiển, hoạt động như một bộ sưu tập các hình ảnh trực quan.

**7.2.1 Tạo 1 visualization.**

Để tạo trực quan hóa mới, chọn Trực quan hóa từ thanh menu trên cùng, mở trang Visualize mới, sau đó bấm vào nút Trực quan hóa mới trên thanh công cụ.

Tạo một trực quan hóa mới là một quá trình ba bước trên trang **Visualize**:

1. Chọn một loại trực quan.

2. Chọn nguồn dữ liệu (từ tìm kiếm mới hoặc tìm kiếm đã lưu).

3. Định cấu hình các tập hợp (số liệu và nhóm) sẽ được sử dụng để hiển thị trên trang Edit.

**Loại trực quan. (Visualization types)**

Kibana hỗ trợ các loại trực quan:

**• Area chart**

**• Data table**

**• Line chart**

**• Markdown widget**

**• Metric**

**• Pie chart**

**• Tile map**

**• Vertical bar chart**

Trước khi chúng tôi bắt đầu xây dựng trực quan hóa các loại khác nhau, chúng ta hãy hiểu một chút về các tập hợp Elaticsearch, tạo thành xương sống của các trực quan hóa trong Kibana.

**+) Số liệu và tổng hợp buckets**.

Các khái niệm số liệu và nhóm đến từ chức năng tổng hợp của Elaticsearch và chúng đóng một vai trò quan trọng khi thiết kế trực quan hóa cho tập dữ liệu của bạn trong Kibana.

**++)Buckets**

Các bucket phân phối tài liệu giữa nhiều nhóm chứa một tập hợp các tài liệu được lập chỉ mục. Các nhóm rất giống với chức năng GROUP BY trong SQL. Họ giúp nhóm các tài liệu dựa trên các tiêu chí được chỉ định và số liệu có thể được áp dụng trên các tài liệu này.

Các bucket thường đại diện cho **X-axis** trong biểu đồ Kibana và có thể thêm các nhóm phụ vào một nhóm.

Các thùng sau đây có sẵn cho **X-axis** trong Kibana:

**• Date Histogram**

**• Histogram**

**• Range**

**• Date Range**

**• IPV4 Range**

**• Terms**

**• Filters**

**• Significant Terms**

**++) Metrics**

Metric biểu thị các tính toán được thực hiện trên các giá trị của các trường trong mỗi nhóm, ví dụ: tính toán số lượng, trung bình, tối thiểu hoặc tối đa của một trường trong tài liệu. Số liệu thường đại diện cho **Y-axis** trong **biểu đồ Area, biểu đồ thanh dọc** và **biểu đồ đường**.

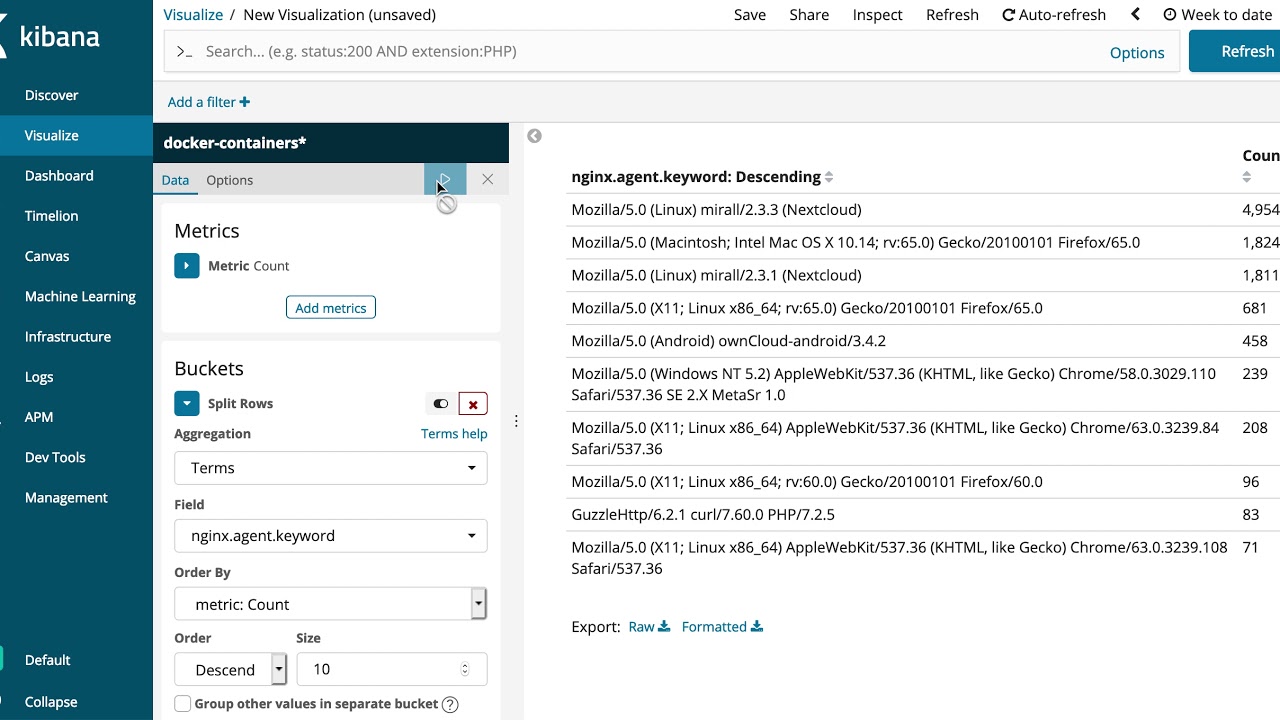
**+) Area Chart.**

Biểu đồ khu vực đặc biệt hữu ích để tạo các mốc thời gian xếp chồng hoặc phân phối dữ liệu.

Biểu đồ khu vực sử dụng các metric làm **Y-axis** và **bucket** cho **X-axis**. Chúng tôi cũng có thể xác định các tập hợp con trong các nhóm, cung cấp cho bạn chức năng của **Biểu đồ phân chia** (nhiều biểu đồ dựa trên các tập hợp khác nhau) hoặc **Khu vực phân chia** (Phân chia biểu đồ khu vực dựa trên các tập hợp khác nhau).

**+) Data table.**

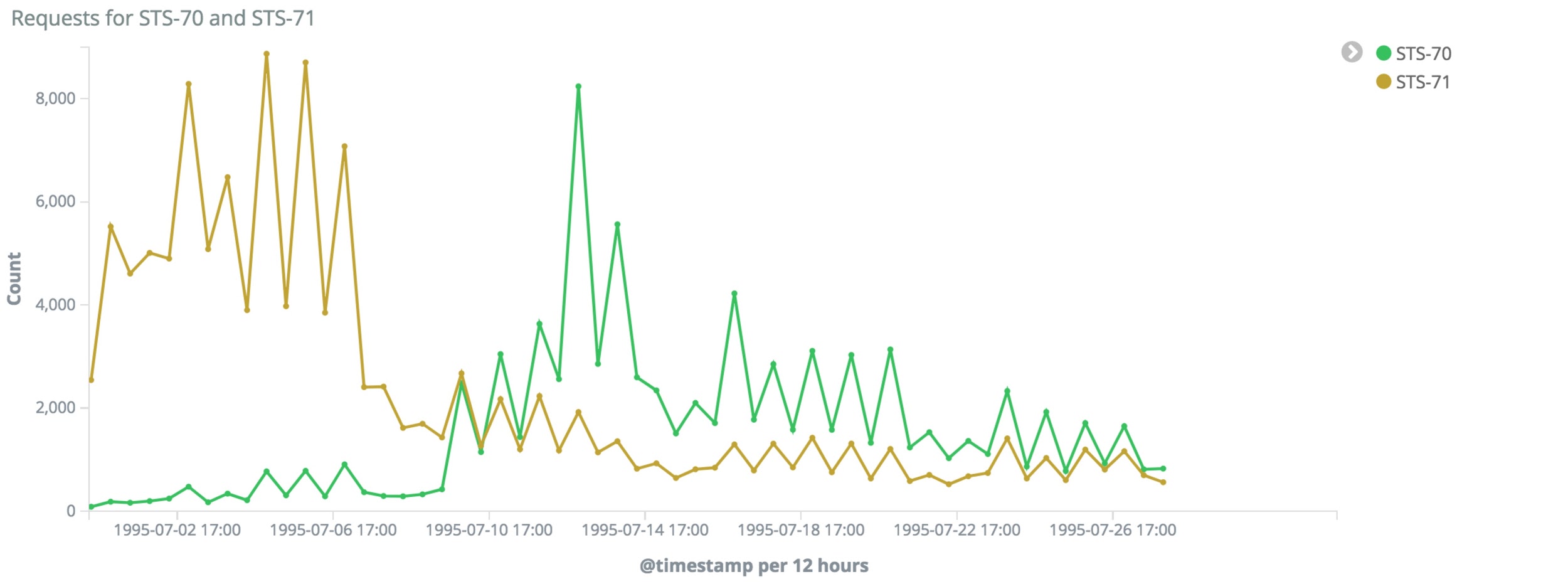
Bảng dữ liệu được sử dụng để trình bày dữ liệu tổng hợp theo định dạng bảng và giúp xác định **N loại tổng hợp hàng đầu.**



Kibana Data Table

**+) Line chart**.

Biểu đồ đường được sử dụng cho chuỗi thời gian mật độ cao và thường hữu ích khi so sánh chuỗi này với chuỗi khác.



Kibana 7 Line chart

**+) Markdown widget.**

Markdown widget được sử dụng để hiển thị thông tin hoặc hướng dẫn trên **Dashboard** và có thể được sử dụng cho mọi yêu cầu đối với văn bản trên **Dashboard**.

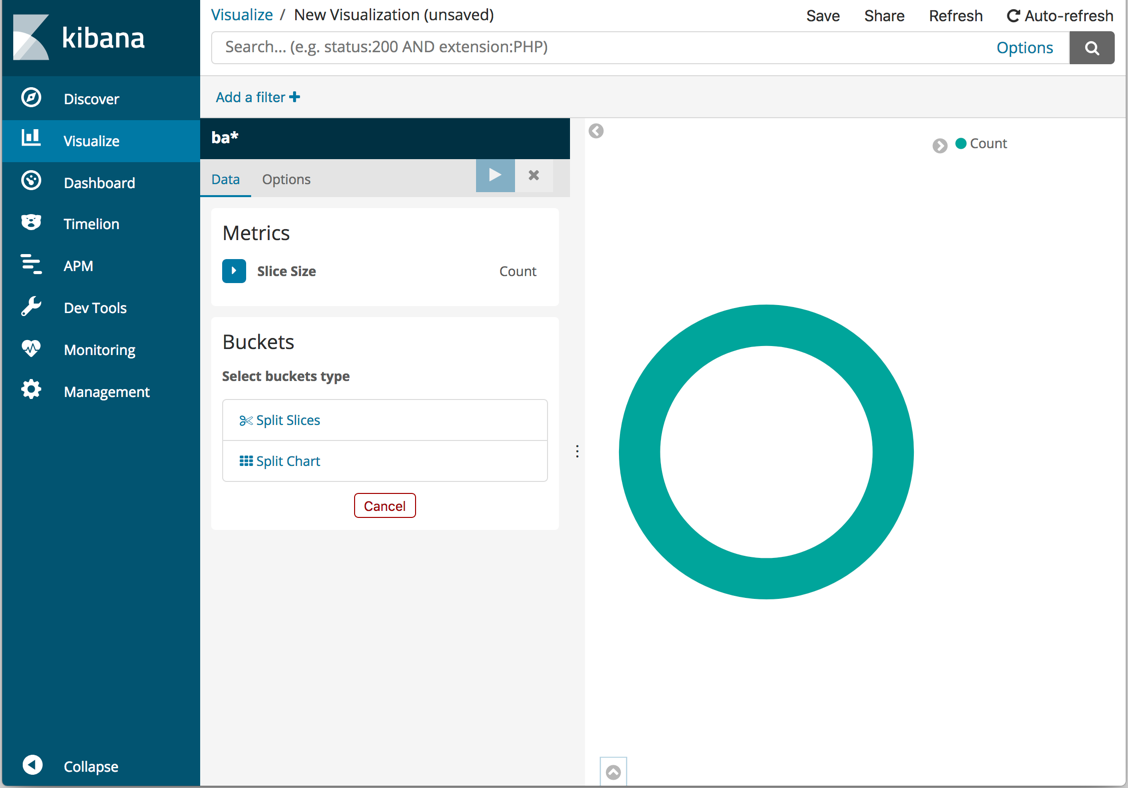
**+) Pipe chart**.

Biểu đồ hình tròn thường được sử dụng để hiển thị các phần của toàn bộ hoặc tỷ lệ phần trăm. Nó đại diện cho việc phân phối dữ liệu qua nhiều lát trong biểu đồ hình tròn.

Một lát của biểu đồ hình tròn được xác định bằng các phép đo tổng hợp, có thể có các giá trị **Count**, **Sum** hoặc **Unique Count**. Tập hợp nhóm xác định loại dữ liệu phải được biểu diễn trong một biểu đồ.

Ví dụ, biểu đồ hình tròn sau đây có thể được sử dụng để hiển thị phân phối của

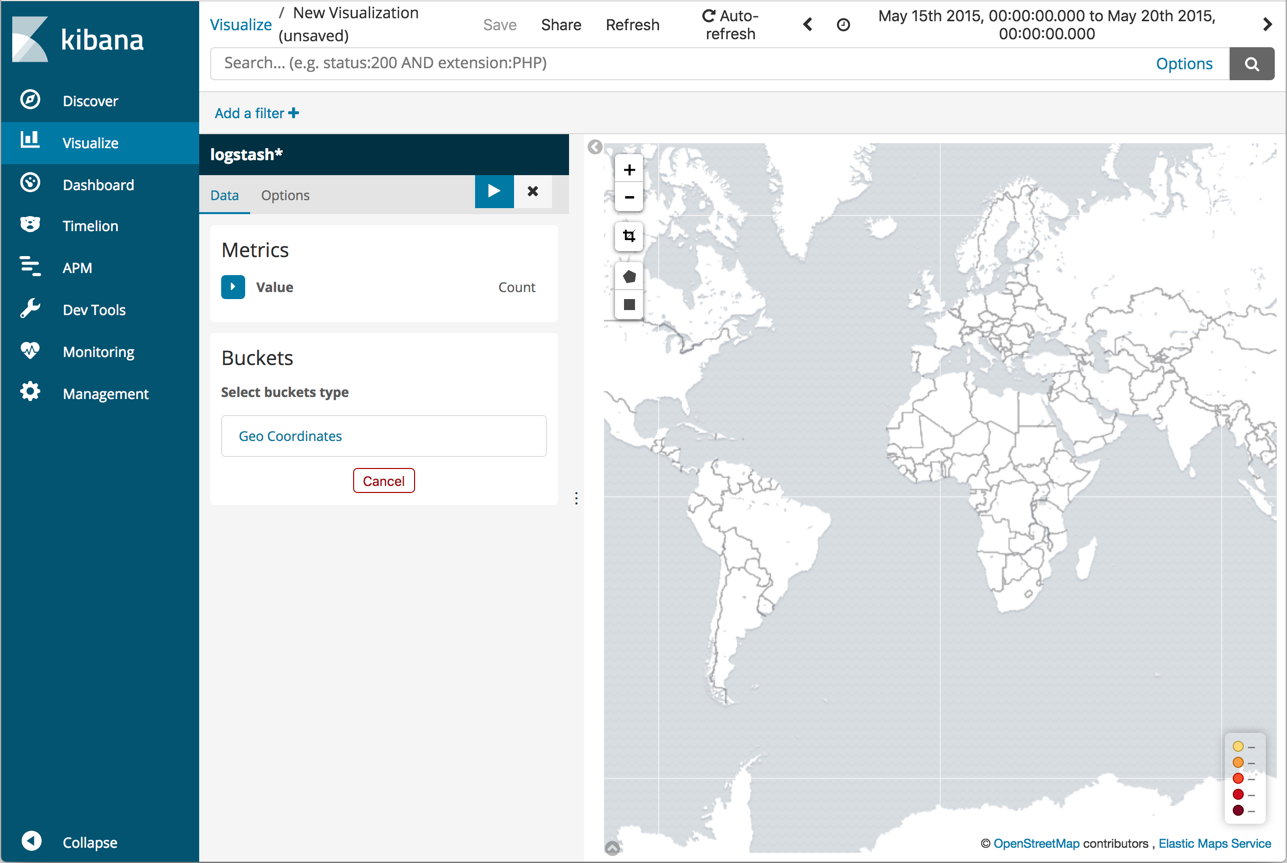
mã phản hồi khác nhau của một ứng dụng.



Kibana Pipe Chart

**+) Tile map**

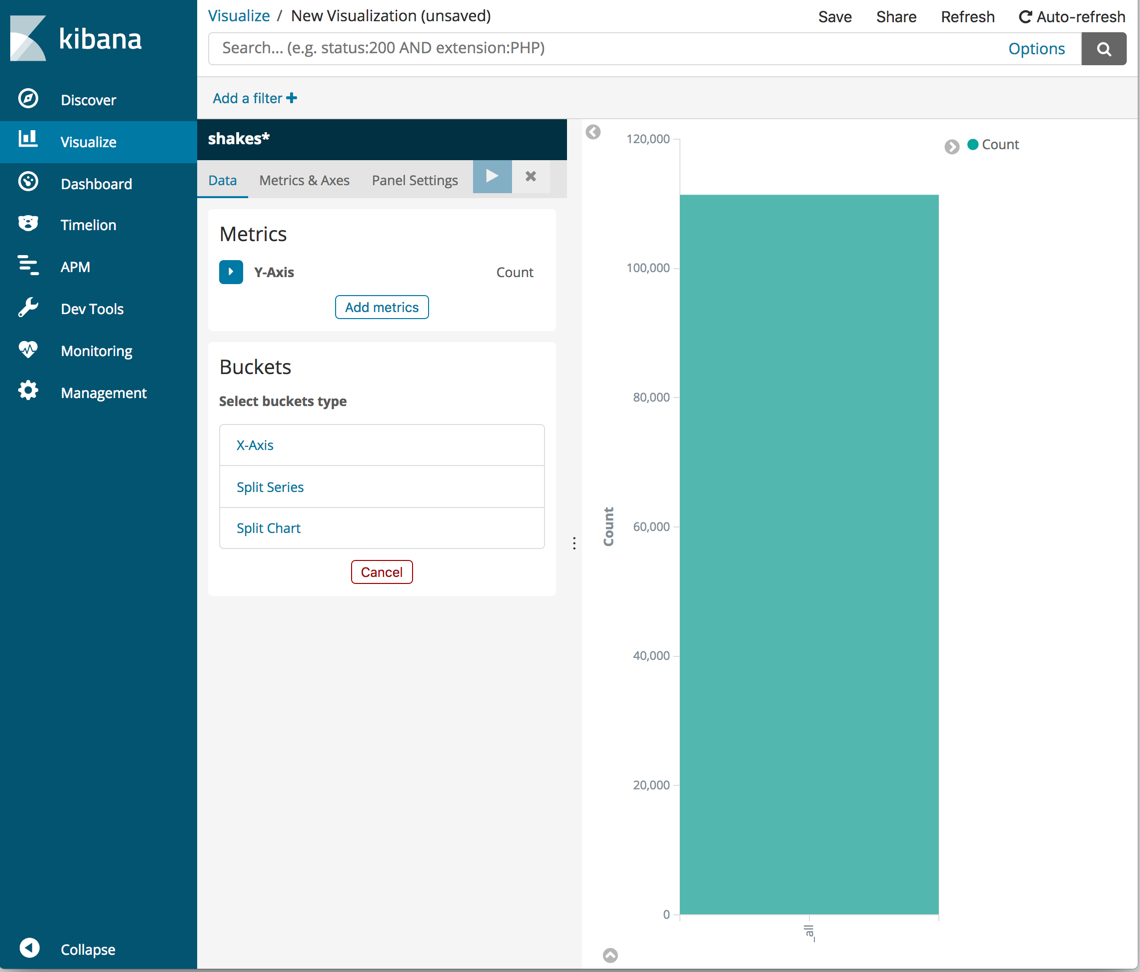
Bản đồ ô vuông được sử dụng để định vị các vị trí địa lý dựa trên tọa độ địa lý. Nó hoạt động trên tập hợp bucket **Geohash**, nhóm nhiều tọa độ thành một nhóm.



Kibana Tile map

**+) Vertical bar chart**.

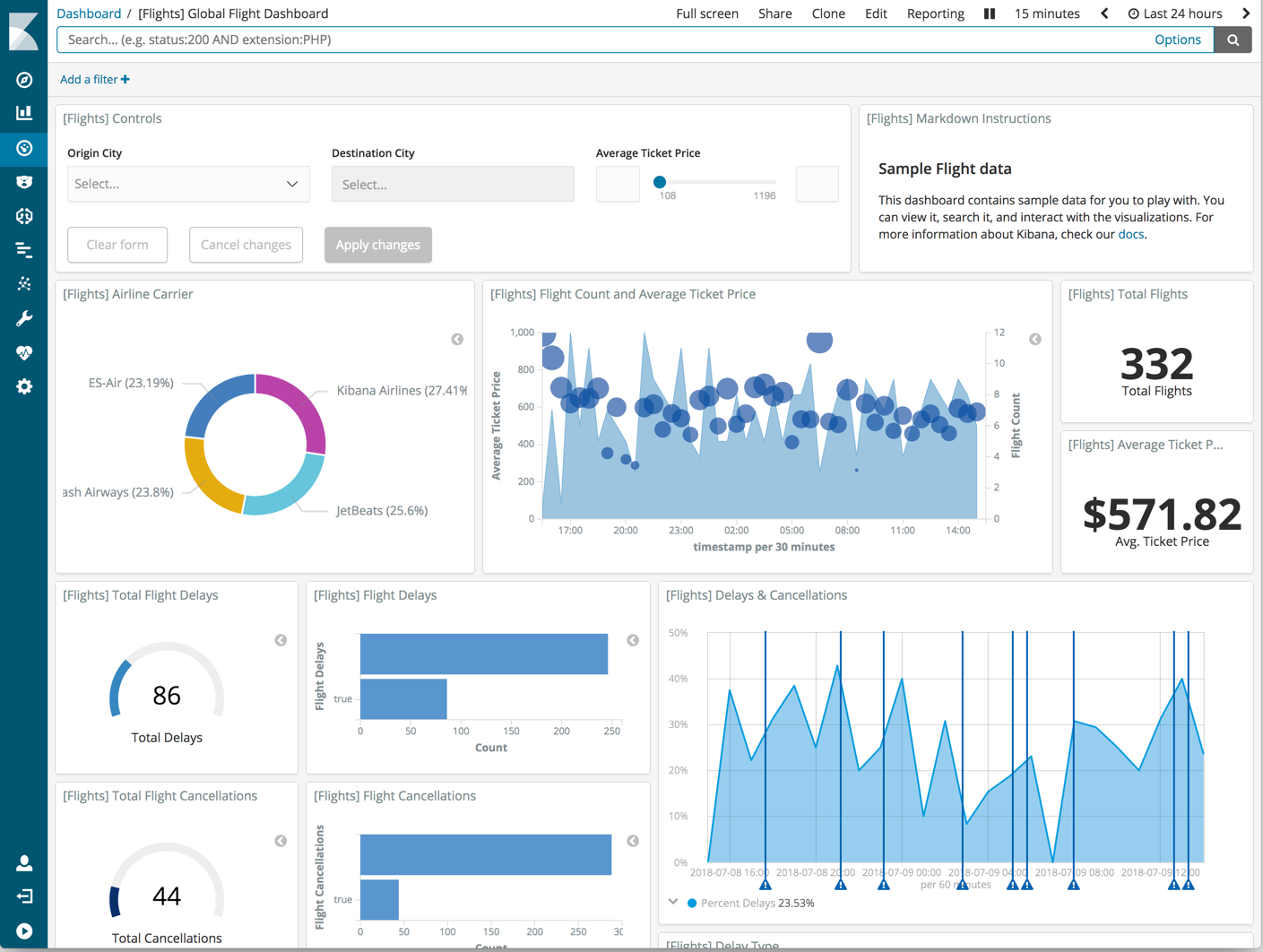
Biểu đồ thanh dọc là biểu đồ có thể được sử dụng cho nhiều mục đích khác nhau và hoạt động tốt với các trường dựa trên thời gian và không dựa trên thời gian. Nó có thể được sử dụng như một thanh đơn hoặc xếp chồng lên nhau. **Y-axis** là số liệu và **X-axis** là **tập hợp bucket.**



Vertical bar chart

**7.3. Dashboard page**.

**Bảng điều khiển** Kibana chỉ là một tập hợp các hình ảnh đã lưu hoặc các tìm kiếm đã lưu, có thể được sắp xếp theo bất kỳ thứ tự nào. Hình ảnh có thể được sử dụng trên nhiều bảng điều khiển và các thay đổi sẽ tự động phản ánh đến tất cả chúng. Một bảng điều khiển có thể được lưu và chia sẻ dễ dàng.

****

Kibana 7 Dashboard

**III. Sự phát triển và lợi ích khi dùng ELK.**

Cho đến nay trong cuốn sách, chúng ta đã thấy cách chúng ta có thể sử dụng ngăn xếp ELK để tìm ra thông tin hữu ích từ nhật ký của mình và xây dựng giải pháp ghi nhật ký tập trung cho nhiều nguồn dữ liệu của một ứng dụng. Trong đường dẫn nhật ký từ đầu đến cuối, chúng tôi đã cấu hình ELK trên máy cục bộ của chúng tôi để sử dụng các phiên bản Elaticsearch, Logstash và Kibana cục bộ.

Trong chương này, chúng ta sẽ xem xét cách ELK Stack có thể được sử dụng trong sản xuất với lượng dữ liệu khổng lồ và nhiều nguồn dữ liệu khác nhau. Một số công ty lớn nhất, như Bloomberg, LinkedIn, Netflix, v.v., đang sử dụng thành công ELK Stack trong sản xuất và ELK Stack đang trở nên phổ biến từng ngày.

Khi chúng tôi nói về việc triển khai cấp độ sản xuất của ELK Stack, một số yêu cầu là:

• Ngăn ngừa mất dữ liệu

• Bảo vệ dữ liệu

• Khả năng mở rộng

• Lưu trữ dữ liệu

**1. Ngăn ngừa mất dữ liệu**.

Ngăn ngừa mất dữ liệu là rất quan trọng đối với một hệ thống sản xuất, vì việc giám sát và gỡ lỗi chủ yếu phụ thuộc vào từng log sự kiện có mặt trong hệ thống; nếu không, toàn bộ phân tích hoặc hệ thống gỡ lỗi sẽ thất bại và cuối cùng chúng tôi sẽ mất một số sự kiện quan trọng trong hệ thống của mình.

Mất dữ liệu có thể được ngăn chặn bằng cách sử dụng một nhà môi giới tin nhắn trước các bộ chỉ mục Logstash. Các nhà môi giới tin nhắn, chẳng hạn như Redis, tỏ ra hữu ích khi xử lý một luồng dữ liệu lớn, vì Logstash có thể làm chậm trong khi lập chỉ mục dữ liệu cho Elaticsearch. Redis có thể giúp đỡ trong những tình huống này, nơi nó có thể đệm dữ liệu trong khi Logstash đang bận lập chỉ mục cho Elaticsearch. Nó cũng thêm một lớp khả năng phục hồi trong đó nếu lập chỉ mục thất bại, các sự kiện được tổ chức trong một hàng đợi thay vì bị mất. ZeroMQ, RabbitMQ, AMQP cũng có thể được sử dụng làm nhà môi giới thay cho Redis

**2. Bảo vệ dữ liệu**.

Vì dữ liệu có giá trị to lớn và mang nhiều thông tin bí mật, nên việc bảo vệ dữ liệu tại các điểm khác nhau trong ELK Stack là vô cùng quan trọng. Các chỉ số Elaticsearch phải được ngăn chặn khỏi sự truy cập trái phép và Bảng điều khiển Kibana cũng cần được bảo vệ. Chúng tôi cũng có thể thiết lập proxy ngược Nginx để truy cập các phiên bản Kibana, điều này sẽ đặt bảng điều khiển Kibana của bạn phía sau trang xác thực yêu cầu tên người dùng và mật khẩu.

Kibana hỗ trợ mã hóa SSL cho cả yêu cầu của khách hàng và các yêu cầu mà máy chủ Kibana gửi tới Elaticsearch.

Để mã hóa thông tin liên lạc giữa trình duyệt và máy chủ Kibana, chúng ta có thể

cấu hình các thuộc tính ssl\_key\_file và ssl\_cert\_file trong kibana.yml:

Sau đây là SSL cho các yêu cầu gửi đi từ máy chủ Kibana

(PEM được định dạng):

• ssl\_key\_file: /path/to/your/server.key

• ssl\_cert\_file: /path/to/your/server.crt

**3. Mở rộng hệ thống**.

Khi dữ liệu trong ứng dụng phát triển, điều cần thiết là hệ thống phân tích nhật ký phải mở rộng tốt với hệ thống. Ngoài ra, đôi khi các hệ thống của bạn chịu tải nặng và bạn cần các hệ thống phân tích nhật ký của mình để phân tích những gì đang xảy ra với ứng dụng. ELK Stack cung cấp khả năng đó nơi bạn có thể dễ dàng chia tỷ lệ từng thành phần theo nhu cầu của mình. Bạn luôn có thể thêm nhiều nút Elaticsearch (nút chính và nút dữ liệu) trong cụm. Bạn nên có ba nút chính (một chính và hai dự phòng) cho các cụm lớn. Ngoài ra, các nút cân bằng tải hoặc định tuyến có thể được thêm vào cho các yêu cầu lập chỉ mục và tìm kiếm khối lượng lớn. Bạn cũng có thể nhận thêm các phiên bản Logstash và Redis và thêm nhiều phiên bản Kibana nữa.

**4. Lưu trữ dữ liệu.**

Khi thiết lập hệ thống phân tích nhật ký, điều cực kỳ quan trọng là xác định chính sách lưu giữ dữ liệu của bạn vì Elaticsearch không thể giữ tất cả dữ liệu bạn có, điều này có thể dẫn đến mất dữ liệu. Cần có một quy trình để tự động xóa các chỉ số cũ sau một khoảng thời gian xác định.

Trình quản lý Elaticsearch (https://github.com/elSTERearch/curator) đặc biệt hữu ích để quản lý các chỉ số của bạn. Bạn có thể lên lịch Curator để xóa các chỉ số cũ dựa trên nhu cầu của bạn. Ví dụ: lệnh sau có thể được thiết lập trong crontab để xóa các chỉ mục cũ hơn 10 ngày tại một thời điểm cụ thể, hàng ngày:

curator --host 10.0.0.7 delete indices --older-than 10 --time-unit days \ --timestring '%Y.%m.%d'

Chú thích một chút: Kibana của tao đang bị lỗi nên hiện tại chưa lên được, nên có 1 số phần bị thiếu hình ảnh để trực quan hơn. Tao nộp bản docs và đang tiếp tục chỉnh sửa).

Tài liệu tham khảo: <https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elastic-stack-on-ubuntu-18-04>

Ebook: Learning ELK Stack.pdf