

Ghi chép về ELK

58 commits

1 branch

0 releases

1 contributor

Branch: master


New pull request

Create new file















Upload files

Find file

Clone or download

 TrongTan124 update

Latest commit 8ca3e15 on Apr 21

 images	update	9 months ago
 scripts	update	6 months ago
 ELK-product.md	update	4 months ago
 Elasticsearch-advance.md	update	9 months ago
 README.md	update	8 months ago
 cai-dat-ELK.md	update	9 months ago
 phan-tich-log-OpenStack.md	update	8 months ago
 read-Elasticsearch-the-definitive-guide.md	update	9 months ago
 tim-hieu-ve-ELK.md	update	9 months ago
 tim-hieu-ve-Elasticsearch.md	update	6 months ago
 tim-hieu-ve-Grok.md	update	9 months ago
 tim-hieu-ve-Kibana.md	update	9 months ago
 tim-hieu-ve-Logstash.md	update	9 months ago
 tim-hieu-ve-beat.md	update	6 months ago

README.md

# Đây là ghi chép của tôi trong quá trình tìm hiểu về ELK (Elasticsearch-Logstash-Kibana)

Có thể những ghi chép này được cóp nhặt vụn vặt từ nhiều nguồn, tôi sẽ để lại link gốc những nguồn mà tôi tham khảo.

ELK là một ứng dụng được tạo lên bằng cách kết hợp các thành phần xử lý khác nhau:

- Elasticsearch: lưu trữ và đánh chỉ mục dữ liệu
- Logstash: Tập trung và phân tích Log
- Kibana: Hiển thị truy vấn

Điểm mạnh của ELK là nó có thể cung cấp kết quả tìm kiếm theo thời gian thực với một lượng dữ liệu cực lớn.

Để hiểu về ELK cần hiểu về cách làm việc của từng thành phần, tôi sẽ tìm hiểu từng thành phần của ứng dụng này.

## 1. Tổng quan

Dữ liệu được các Beats (shipper) thu thập và gửi về cho Logstash; Logstash tiếp nhận và phân tích dữ liệu. Sau đó dữ liệu được gửi vào Elasticsearch; Elasticsearch nhận dữ liệu từ Logstash và lưu trữ, đánh chỉ mục; Kibana sử dụng các dữ liệu trong Elasticsearch để hiển thị và phân tích cú pháp tìm kiếm mà người dùng nhập vào để gửi cho Elasticsearch tìm kiếm.

Beats --> Logstash --> Elasticsearch <--> Kibana

## 2. Beats

---

Được viết bằng Golang, chạy trên các client để thu thập dữ liệu. Bạn có thể sử dụng libbeat để viết các beats cho mục đích thu thập của riêng mình

Hiện tại có các Beats được cung cấp sẵn bởi elastic là:

- Filebeat: đọc file và lưu vị trí cuối cùng, khi có dữ liệu mới sẽ đọc tiếp và gửi
- Packetbeat: capture gói tin trên các port của client, chuyển tiếp dữ liệu về Logstash
- Topbeat (metricbeat): Thu thập dữ liệu về tài nguyên hệ thống client và gửi về Logstash
- Winlogbeat: Thu thập dữ liệu trên windows (Chưa thử, nhưng có thể sử dụng nxlog để thu thập)

## 3. Logstash

---

Nhận dữ liệu từ các beats, tiến hành phân tích dữ liệu

Phân tích dữ liệu gửi từ filebeat bằng grok

Grok là một dạng khai báo pattern sử dụng regular expression

Grok đã được khai báo rất nhiều pattern có sẵn, bạn có thể sử dụng ngay.

Nếu bạn có các loại dữ liệu đặc thù thì có thể dựa vào regular expression để khai báo các pattern theo yêu cầu

Các dữ liệu packetbeat, topbeat thì có các template sẵn nên không cần khai báo filter, được logstash gửi thẳng vào Elasticsearch

Dữ liệu được gửi sang cho Elasticsearch để lưu trữ

## 4. Elasticsearch

---

Thực hiện lưu trữ và đánh chỉ mục dữ liệu.

Sử dụng các template để lưu trữ dữ liệu

Có thể cấu hình cluster, shard, replica để tăng tính an toàn, tính sẵn sàng, tăng hiệu năng đánh chỉ mục, tăng hiệu năng tìm kiếm dữ liệu (phần này nâng cao)

## 5. Kibana

---

Hiển thị dữ liệu theo thời gian thực

Hỗ trợ tìm kiếm dữ liệu theo nhiều kiểu

Hiển thị dữ liệu theo nhiều dạng biểu đồ

## 6. Nâng cao

---

ELK được thử nghiệm là mô hình AIO (all in one), nhưng để thành một sản phẩm đạt tiêu chuẩn thương mại:

- Ổn định
- an toàn
- nhanh chóng

thì cần một số thiết lập riêng như sau:

- Sử dụng Redis để nhận dữ liệu từ các beats
- Chuyển tiếp dữ liệu từ Redis vào Logstash
- Cài đặt Redis và Logstash tại một node riêng
- Cài đặt Elasticsearch trên 02 node và khai báo replica là 1, shard có thể để mặc định là 5 hoặc tăng lên tùy theo nhu cầu

- Kibana nên cài đặt cùng nginx trên một node riêng để hỗ trợ xác thực user đăng nhập và hiển thị.
- Sử dụng IP VIP cho Elasticsearch.
- Optimized Elasticsearch để tăng hiệu năng: limit open file, heap size process, read write disk (xài SSD và đặt raid0 để truy vấn), memory (cache, calculate index file)
- Còn yêu cầu nào khác sẽ bổ sung tiếp :)

## 7. Q&A

- Ngoài ELK stack còn sản phẩm nào tương tự không?
  - Có nhiều. free hoặc thương mại đều có. splunk, graylog
- Tại sao ELK lại dùng Elasticsearch mà không phải ứng dụng lưu trữ khác như MongoDB, solr,...?
  - Elasticsearch giống solr, đều là một search engine, tức là nó hỗ trợ tìm kiếm full keyword. Sử dụng cluster, shard, replica để phân tán dữ liệu và tăng hiệu năng lưu trữ, tìm kiếm.
  - Tại sao dùng Elasticsearch mà ko phải solr thì chưa rõ. Có google ở [đây](#) :))
- Các beats chạy có tiêu tốn tài nguyên hệ thống không?
  - Beats được viết bằng Golang, là một ngôn ngữ lập trình quản lý bộ nhớ rất tốt, nên lượng tài nguyên dành cho beats hoàn toàn ko đáng kể. Nhưng nếu chạy packetbeat mà lượng packet capture cao thì cũng cần kiểm tra lại. :)
- Có câu hỏi nào bạn cứ post để cùng tìm câu trả lời. :)
- Rất mong được học hỏi từ tất cả mọi người.

Mọi ý kiến đóng góp có thể phản hồi theo địa chỉ sau:

- Skype: crazyman12487
- Gmail: [nguyentrongtan124@gmail.com](mailto:nguyentrongtan124@gmail.com)

