TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

----- 🙡 🕮 🙣 -----



**BÁO CÁO PROJECT III**

***Tuần 1:***

***Tìm hiểu về tập tin log và bộ công cụ ELK***

Giảng viên: **TS. Nguyễn Đức Toàn**

|  |  |
| --- | --- |
| Sinh viên thực hiện: | **Lê Hà Hưng** |
| MSSV: | **20183757** |

**Hà Nội, tháng 10 năm 2021**

Mục lục

[1. Tìm hiểu về tập tin log 3](#_Toc85281740)

[ **Khái niệm:** 3](#_Toc85281741)

[ **Phân loại log**: 3](#_Toc85281742)

[2. Tìm hiểu về bộ công cụ ELK (Elasticsearch, Logstash, Kibana) 5](#_Toc85281743)

[**2.1.** **Tìm hiểu về Elasticsearch** 5](#_Toc85281744)

[**2.2.** **Tìm hiểu về Logstash** 7](#_Toc85281745)

[**2.3.** **Tìm hiểu về Kibana** 10](#_Toc85281746)

[**2.4.** **Hệ sinh thái ELK STACK** 11](#_Toc85281747)

1. **Tìm hiểu về tập tin log**

* **Khái niệm:**

Log file là một tệp nhật ký dữ liệu được sử dụng bởi tất cả các phần mềm và hệ điều hành. Log file được dùng để có thể xem lại các thông tin hoạt động của ứng dụng trong quá khứ như debug khi có lỗi xảy ra, check health, xem info, error, warning, …

* **Ví dụ về log:**

Khi xử lý một HTTP request từ phía client, request này khi được xử lý thì gây ra lỗi 500 – “Internal server error”. Khi đó thông tin log ở đây ít nhất phải có:

* + Thời gian request.
  + Người request.
  + HTTP request info: header, request, body, …
  + HTTP response info.
  + Error stack trace về error đó như lỗi ở đoạn nào, dòng nào, lỗi gì, input như thế nào, …

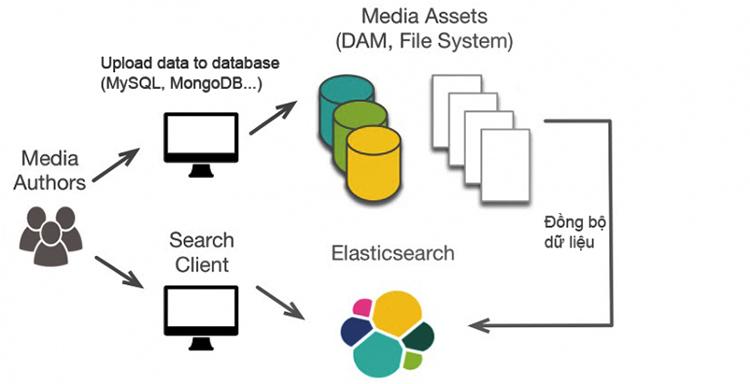
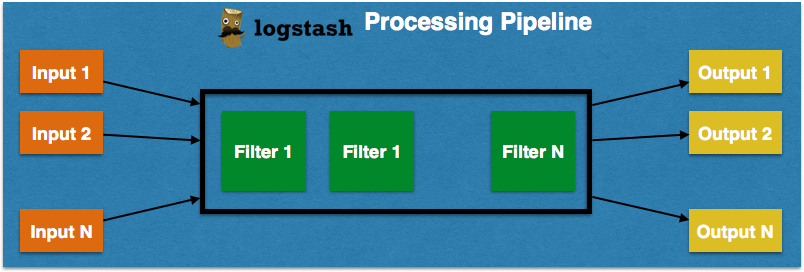
Khi xem lại đoạn log, chúng ta biết được cách tái hiện lại lỗi hay phán đoán lỗi xảy ra như thế nào để khác phục nhanh hơn và chính xác hơn.

* **Phân loại log**:

Phân loại theo level tuỳ theo mục đích sử dụng.

* + **All:** cấp độ thấp nhất, Logger (Chịu trách nhiệm thu thập thông tin log) và Appender (Chịu trách nhiệm ghi log tới các vị trí đã được cấu hình (file, console)) được định nghĩa với cấp độ này, mọi thông tin cần log sẽ được log.
  + **Debug:** các thông tin dùng để debug, chúng ta có thể bật/ tắt log này dựa vào mode của application.
  + **Info:** các thông tin muốn ghi nhận thêm trong quá trình hoạt động của hệ thống. Ví dụ: log số lượng request, status, … để biết lượng truy cập của hệ thống thế nào.
  + **Warning:** log các thông tin cảnh báo của chương trình.
  + **Error:** các lỗi khi chạy chương trình sẽ được log.
  + **Fatal:** log các lỗi nghiêm trọng xảy ra trong chương trình, có thể làm cho chương trình không sử dụng được nữa.
  + **Off**: đây là cấp độ cao nhất, được sử dụng khi chúng ta không muốn log bất kỳ thông tin nào nữa.
* Log format:
  + Log đó phải sử dụng được, nghĩa là dễ đọc, dễ trace ra được thông tin cần tìm. Để làm được điều này, các log của chúng ta phải tuân thủ theo một format được chỉ định.
  + Một trong những ứng dụng trong Java thường dùng là Logstash. Vì vậy file log nên tuân thủ theo format của các ứng dụng này.
* Log rotate: log rotate là việc cắt nhỏ log ra và lưu trữ trên nhiều file thay vì một file.
  + Có thể sẽ lưu file log riêng theo từng ngày, tuần hoặc tháng.
  + File log sẽ chia theo loại log
  + …

1. **Tìm hiểu về bộ công cụ ELK (Elasticsearch, Logstash, Kibana)**
   1. **Tìm hiểu về Elasticsearch**

* **Định nghĩa:** 
  + Theo Wikipedia, Elasticsearch (ES) được định nghĩa là một công cụ tìm kiếm dựa vào nền tảng Apache Lucene (phần mềm mã nguồn mở, dùng để phân tích, đánh chỉ mục và tìm kiếm thông tin với hiệu suất cao bằng Java). Công cụ này cung cấp một bộ máy tìm kiếm có dạng phân tán với đầy đủ công cụ với một giao diện web HTTP có hỗ trợ cho các dữ liệu JSON
  + Elasticsearch được phát triển bằng Java và được phát hành dạng nguồn mở theo giấy phép Apache
  + Elasticsearch hoạt động thực chất sẽ như 1 web server với khả năng tìm kiếm nhanh chóng thông qua giao thức RESTful
  + Elasticsearch chạy trên server riêng và đồng thời giao tiếp thông qua RESTful do vậy nên nó không phụ thuộc vào client viết bằng gì hay hệ thống hiện tại viết bằng gì. Nên việc tích hợp nó vào hệ thống là dễ dàng, chỉ cần gửi request http lên là nó trả về kết quả
  + Elasticsearch là một hệ thống phân tán với khả năng mở rộng vô cùng tuyệt vời. Chỉ cần lắp thêm node cho nó là nó sẽ auto mở rộng
* **Cách thức hoạt động của Elasticsearch:**
  + Hoạt động của Elasticsearch được hiểu là một server riêng biệt giúp “phục vụ” các nhu cầu tìm kiếm dữ liệu
  + ES mặc dù không quá mạnh trong thao tác CRUD nên thường được sử dụng song song với 1 DB chính (SQL, mySQL, MongoDB, …)
* **Các khái niệm khác liên quan đến Elasticsearch:**
  + **Document trong Elasticsearch:** Document là một JSON object với một số dữ liệu riêng, đây là basic information unit có trong ES. Đây chính là một đơn vị nhỏ nhất giúp lưu trữ các dữ liệu có trong Elasticsearch
  + **Index:** 
    - Trong Elasticsearch việc sử dụng một cấu trúc sẽ được gọi là inverted index và nó thường được thiết kế để chúng ta có thể tìm kiếm full-text search
    - Cách thức của nó tương đối đơn giản và các văn bản đều sẽ được phân tách ra thành những từng có ý nghĩa, sau đó sẽ được map tìm hiểu xem nó thuộc dạng văn bản nào. Khi đó, search sẽ tùy thuộc vào loại search để có thể đưa ra được kết quả cuối cùng
  + **Shard:** 
    - Là đối tượng của Lucene, là tập con các documents của 1 Index. Một Index có thể được chia thành nhiều shard
    - Mỗi node bao gồm nhiều Shard. Chính vì thế Shard mà là đối tượng nhỏ nhất, hoạt động ở mức thấp nhất, đóng vai trò lưu trữ dữ liệu
    - Có 2 loại shard là Primary shard và Replica shard:
    - **Primary Shard:** có nhiệm vụ lưu dữ liệu và đánh index, sau khi các dữ liệu đã được đánh xong thì sẽ được vận chuyển tới các Replica Shard. Mặc định của Elasticsearch chính là mỗi index sẽ có 5 primary shard và với một Primary Shard thì nó sẽ đi kèm với 1 Replica Shard
    - **Replica Shard:** nó chính là sự lưu trữ dữ liệu nhân bản của Primary Shard. Replica Shard có vai trò quan trọng nhằm đảm bảo tính toàn vẹn cho dữ liệu trong trường hợp Replica Shard xảy ra vấn đề. Ngoài ra nó còn giúp tăng tốc độ tìm kiếm
  + **Node:** 
    - Là trung tâm hoạt động của Elasticsearch, là nơi giữ nhiệm vụ lưu trữ dữ liệu, tham gia vào thực hiện đánh index của cluster cũng như thực hiện những thao tác tìm kiếm khác.
    - Mỗi node thường được định danh bằng 1 unique name.
  + **Cluster:**
    - Là tập hợp các nodes đang hoạt động chung với nhau và chia sẻ cùng thuộc tính cluster.name
    - Cluster sẽ được xác định bằng 1 “unique name”
    - Chức năng chính của Cluster đó chính là quyết định xem shards nào được phân bổ cho node nào và khi nào thì di chuyển các Cluster để cân bằng lại Cluster
    - Mỗi cluster có một node chính (master), được lựa chọn một cách tự động và có thể thay thế nếu sự cố xảy ra
    - Một cluster có thể gồm 1 hoặc nhiều nodes. Các nodes có thể hoạt động trên cùng 1 server. Thông thường một cluster sẽ gồm nhiều nodes hoạt động trên các server khác nhau để đảm bảo nếu 1 server gặp sự cố thì server khác (node khác) có thể hoạt động đầy đủ chức năng so với khi có 2 servers.
* **Ưu điểm của ES:**
  + Hỗ trợ tìm kiếm dữ liệu mạnh mã, nhanh chóng dựa trên Apache Lucene
  + Mang lại khả năng phân tích dữ liệu vô cùng hiệu quả
  + Có khả năng mở rộng dựa theo chiều ngang rất tuyệt vời
  + Công cụ này có khả năng hỗ trợ tìm kiếm, nếu như từ khóa tìm kiếm bị lỗi chính tả hoặc không đúng cú pháp thì vẫn có khả năng Elasticsearch trả về với kết quả cực tốt
  + Elasticsearch có thể hỗ trợ Structured Query DSL cung cấp cho nhu cầu đặc tả những câu truy vấn phức tạp hơn một cách rõ ràng và cụ thể bằng JSON
  + Elasticsearch hỗ trợ cho nhiều Elasticsearc client như PhP, Java, Ruby, Javascript, Python, .NET, …
* **Nhược điểm:**
  + Được thiết kế cho mục đích search, do vậy với những nhiệm vụ khác ngoài search như CRUD thì elastic kém thế hơn so với những database khác như Mongodb, mySql, …
  + Không có khái niệm database transaction, tức là nó sẽ không đảm bảo được toàn vẹn dữ liệu trong các hoạt động Insert, Update, Delete
  + Không thích hợp với những hệ thống thường xuyên cập nhật dữ liệu. Sẽ rất tốn kém cho việc đánh index dữ liệu
  1. **Tìm hiểu về Logstash**
* **Khái niệm**:
  + Logstash là một công cụ mã nguồn mở thu thập dữ liệu có khả năng liên hợp theo thời gian thực. Logstash có thể hợp nhất dữ liệu từ các nguồn khác nhau và chuẩn hóa dữ liệu ở phần xử lý tiếp theo. Loại bỏ và đồng hóa tất cả dữ liệu đó trong một số use case cần phân tích và thể hiện trên biểu đồ
  + Logstash có 3 thành phần chính cũng chính là 3 bước xử lý chính:
    - **INPUT:** tiếp nhận/thu thập dữ liệu sự kiện log ở dạng thô từ các nguồn khác nhau như file, redis, rabbitmq, beats, syslog, …
    - **FILTER:** Sau khi tiếp nhận dữ liệu sẽ tiến hành thao tác dữ liệu sự kiện log (như thêm, xoá, thay thế, ... nội dung log) theo cấu hình của quản trị viên để xây dựng lại cấu trúc dữ liệu log event theo mong muốn
    - **OUTPUT**: Sau cùng sẽ thực hiện chuyển tiếp dữ liệu sự kiện log về các dịch vụ khác như Elasticsearch tiếp nhận lưu trữ log hoặc hiển thị log, ...
* **Hoạt động của Logstash:**
  + Đường ống xử lý sự kiện của Logstash có ba giai đoạn: input → filter → output. Các đầu vào tạo ra các sự kiện, bộ lọc sửa đổi chúng và các đầu ra sẽ chuyển chúng tới nơi khác
  + Ở bước INPUT, Logstash sẽ được cấu hình lựa chọn hình thức tiếp nhận log event hoặc đi lấy dữ liệu log ở dịch remote theo nhu cầu. Sau khi lấy được log event thì, khâu INPUT sẽ ghi dữ liệu event xuống hàng đợi tập trung ở bộ nhớ RAM hoặc trên ổ cứng.
  + Mỗi pipeline worker thread sẽ tiếp tục lấy một loạt sự kiện đang nằm trong hàng đợi này để xử lý FILTER giúp tái cấu trúc dữ liệu log sẽ được gửi đi ở phần OUTPUT.
* **INPUT:** Một số đầu vào thường được sử dụng là:
  + File: đọc từ một tệp trên hệ thống
  + Syslog: chương trình Logstash sẽ lắng nghe trên port 514 để tiếp nhận dữ liệu syslog
  + Redis: đọc dữ liệu log từ redis server, sử dụng cả 2 cơ chế redis channel và redis lists
  + A picture containing diagram

    Description automatically generatedBeats: xử lý các dữ liệu thông tin được gửi từ chương trình Beats
* **FILTER:** Có thể kết hợp **filter** với các điều kiện so sánh nhằm thực hiện 1 tác vụ hành động (action) khi một sự kiện thoả mãn khớp với các tiêu chí được ra. Một số **filter plugin** hữu ích:
  + **grok:** Nếu bắt gặp một dữ liệu sự kiện log với cấu trúc văn bản không phổ biến hoặc là phức tạp, thì Grok hiện là plugin filter tốt nhất để phân tích cú pháp dữ liệu log không được cấu trúc văn bản thành một thứ có cấu trúc và có thể truy vấn được.
  + **mutate:** thực hiện sự thay đổi trên thông tin sự kiện log như: đổi tên, xoá, thay thế, tinh chỉnh các trường (field) thông tin của sự kiện log.
  + **drop:** dừng xử lý sự kiện ngay lập tức, ví dụ các ‘debug event’.
  + **clone:** tạo một bản copy của sự kiện.
  + **geoip:** thêm thông tin về vị trí địa lý của địa chỉ IP (thường để hiển thị biểu đồ trên Kibana)

Graphical user interface, text, application, chat or text message

Description automatically generated

* **OUTPUT:** là bước cuối cùng trong chuỗi các bước xử lý của Logstash. Một sự kiện có thể đưa qua nhiều output khác nhau. Các output plugin thường sử dụng:
  + **Elasticsearch:** gửi dữ liệu sự kiện đến hệ thống Elasticsearch
  + **File:** có thể lưu output ra file trên hệ thống.
  + **Graphite:** gửi dữ liệu tới graphite, một trong những tool mã nguồn mở hỗ trợ việc lưu trữ và tạo biểu đồ metric.
  + **Statsd:** gửi dữ liệu tới dịch vụ ‘statsd’
  1. **Tìm hiểu về Kibana**
* **Khái niệm:** Là một ứng dụng giao diện người dùng mở và miễn phí nằm trên Elastic Stack, cung cấp khả năng tìm kiếm và trực quan hóa dữ liệu cho dữ liệu được lập chỉ mục trong Elasticsearch.
* **Kibana được sử dụng để:** 
  + Tìm kiếm, xem và trực quan hóa dữ liệu được lập chỉ mục trong Elasticsearch và phân tích dữ liệu thông qua việc tạo biểu đồ thanh, biểu đồ hình tròn, bảng, biểu đồ và bản đồ.
  + Chế độ xem bảng điều khiển kết hợp các yếu tố hình ảnh này sau đó được chia sẻ qua trình duyệt để cung cấp chế độ xem phân tích thời gian thực thành khối lượng dữ liệu lớn để hỗ trợ các trường hợp sử dụng như:
    - Ghi nhật ký và phân tích nhật ký
    - Các chỉ số cơ sở hạ tầng và giám sát container
    - Giám sát hiệu suất ứng dụng (APM)
    - Phân tích và hiển thị dữ liệu không gian địa lý
    - Phân tích bảo mật
    - Phân tích kinh doanh
  + Giám sát, quản lý và bảo mật một phiên bản Elastic Stack thông qua giao diện web.
  + Tập trung quyền truy cập cho các giải pháp tích hợp được phát triển trên Elastic Stack cho khả năng quan sát, bảo mật và các ứng dụng tìm kiếm doanh nghiệp
  1. **Hệ sinh thái ELK STACK**
* **Khái niệm**: ELK Stack là tập hợp 3 phần mềm đi chung với nhau, phục vụ cho công việc logging. Ba phần mềm này lần lượt là:
  + Elasticsearch: Cơ sở dữ liệu để lưu trữ, tìm kiếm và query log
  + Logstash: Tiếp nhận log từ nhiều nguồn, sau đó xử lý log và ghi dữ liệu và Elasticsearch
  + Kibana: Giao diện để quản lý, thống kê log. Đọc thông tin từ Elasticsearch
* Cơ chế hoạt động:
  + Đầu tiên, log sẽ được đưa đến Logstash.
  + Logstash sẽ đọc những log này, thêm những thông tin như thời gian, IP, parse dữ liệu từ log ra, sau đó ghi xuống database là Elasticsearch.
  + Khi muốn xem log, người dùng vào URL của Kibana. Kibana sẽ đọc thông tin log trong Elasticsearch, hiển thị lên giao diện cho người dùng query và xử lý.
* Ưu điểm của ELK Stack:
  + Đọc log từ nhiều nguồn: Logstash có thể đọc được log từ rất nhiều nguồn, từ log file cho đến log database cho đến UDP hay REST request.
  + Dễ tích hợp: Dù chúng ta có dùng MSSQL, MongoDB hay Redis, Logstash đều có thể đọc hiểu và xử lý log của bạn nên việc tích hợp rất dễ dàng.
  + Search và filter mạnh mẽ: Elasticsearch cho phép lưu trữ thông tin kiểu NoSQL, hỗ trợ luôn Full-Text Search nên việc query rất dễ dàng và mạnh mẽ.
  + Hoàn toàn miễn phí.