**Hệ thống quản lý log tập chung ELK stack**

**1. Log là gì**

Logging là một công cụ đơn giản và mạnh mẽ, ghi lại toàn bộ những hoạt động của hệ thống.

Nhờ có logging, ta có thể tra cứu lại trạng thái của hệ thống trong quá khứ, những code nào đã được chạy, từ đó tìm ra lỗi và fix dễ dàng hơn.

**2. Vấn đề của logging**

Trước đây, các hệ thống hay gặp phải vấn đề về logging:

* Hệ thống back-end xây dựng theo mô hình microservice. Mỗi microservice được deploy thành một AppService trên Cloud. Mỗi khi service có vấn đề, cần đọc log thì phải vào Cloud/Server, tìm service đó, tìm trong folder deploy để tìm file log.
* Một app API khác do mình viết bằng Nodejs/Java, cần chạy trên nhiều máy ảo của nhiều server. Mỗi khi app có vấn đề gì thì phải login vào máy ảo đó và tìm file log.
* Khi cần tìm và fix bug, ta cần phải mò và đọc file log để “điều tra” xem lỗi nằm ở đâu.

**3. Tại sao cần hệ thống quản lý log tập chung**

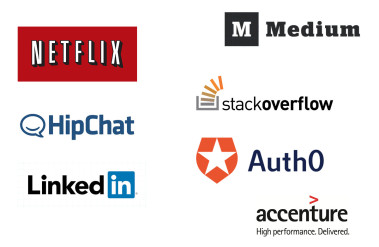
Với các hệ thống hoặc ứng dụng nhỏ, ta không cần sử dụng centralized logging làm gì! Cứ **dùng thư viện ghi log** đi kèm với ngôn ngữ, sau đó ghi log ra file rồi đọc thôi!

Tuy nhiên, với những hệ thống lớn nhiều người dùng, có nhiều service phân tán (microservice), có nhiều server chạy cùng lúc… thì việc ghi log xuống file **không còn hiệu quả nữa**.

Giả sử bạn có 10 con server chạy cùng lúc, bạn sẽ phải **lục tung 10 con server** này để đọc và tìm file log, cực quá phải không nào? Lúc này, người ta bắt đầu áp dụng centralized logging, tức **ghi log tập trung** vào 1 chỗ.

**4. Monitoring log với ELK Stack**

ELK là một lựa chọn phù hợp, được nhiều công ty lớn như **Netflix, Linkedin, Medium** lựa chọn với các lý do sau:



* **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 bạn có dùng Nginx hay Apache, dùng MSSQL, MongoDB hay Redis, Logstash đều có thể đọc hiểu và xử lý log của bạn nên việc tích hợp rất dễ dàng.
* **Hoàn toàn free**: Chỉ cần tải về, setup và dùng, không tốn một đồng nào cả. Công ty tạo ra ELK Stack kiếm tiền bằng các dịch vụ cloud hoặc các sản phẩm premium phụ thêm.
* **Khả năng scale tốt**: Logstash và Elasticsearch chạy trên nhiều node nên hệ thống ELK cực kì dễ scale. Khi có thêm service, thêm người dùng, muốn log nhiều hơn, bạn chỉ việc thêm node cho Logstash và Elasticsearch là xong.
* **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.
* **Cộng đồng mạnh, tutorial nhiều**.

**5. ELK Stack là gì ?**

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 của ELK Stack như sau:



1. Đầu tiên, log sẽ được đưa đến *Logstash*. (Thông qua nhiều con đường, ví dụ như server gửi UDP request chứa log tới URL của Logstash, hoặc Beat đọc file log và gửi lên Logstash)
2. *Logstash* sẽ đọc những log này, thêm những thông tin như thời gian, IP, parse dữ liệu từ log (server nào, độ nghiêm trọng, nội dung log) ra, sau đó ghi xuống database là *Elasticsearch*.
3. 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.

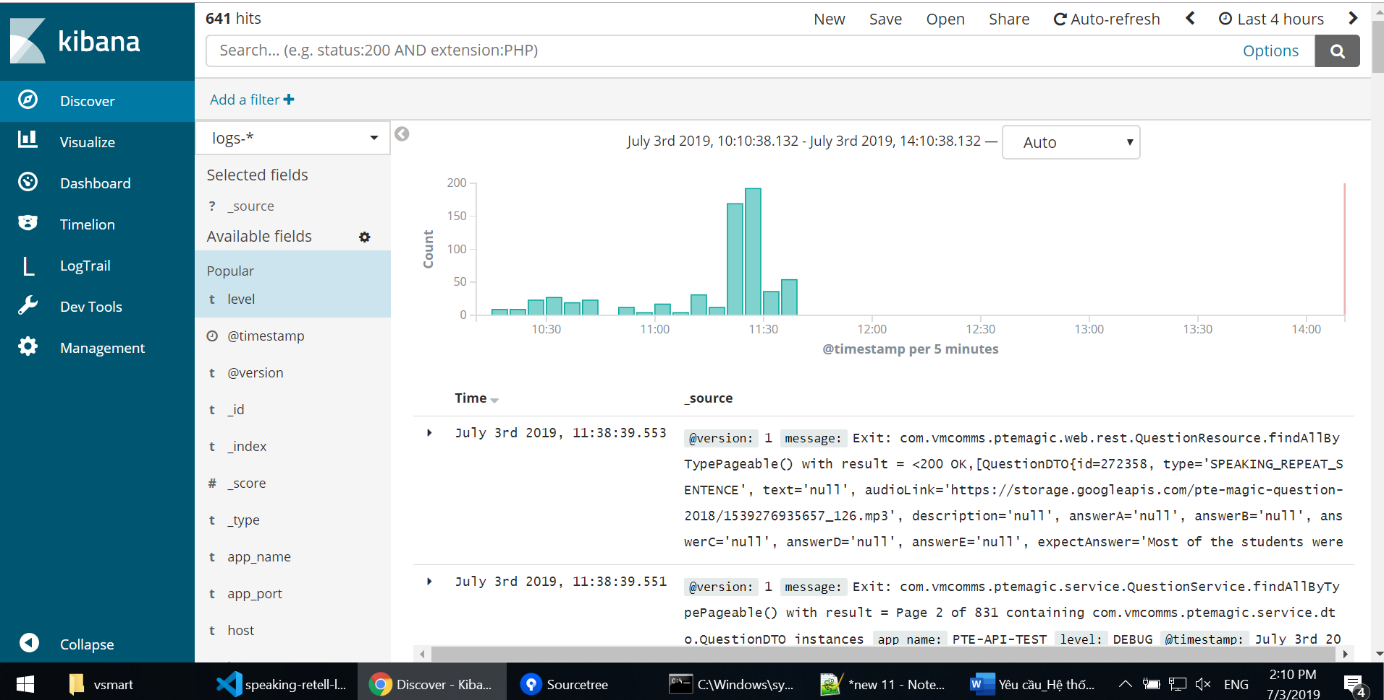
**6. Các bước cần thực hiện với hệ thống hiện tại**

* 1. **Cài đặt ELK Stack server sử dụng Docker**

* Cài ELK Monitoring log lên server Ubuntu 1x.04 LST
* Sử dụng Docker-compose :

# curl -O https://raw.githubusercontent.com/jhipster/jhipster-console/master/bootstrap/docker-compose.yml

docker-compose up -d

.

* 1. **Implement logger sử dụng logstack với Javascript (Reactjs/Nodejs)**

* Sử dụng lib javascript như: winton và Winston-logstash để đẩy log lên server.

let winston = require('winston')

require('winston-logstash');

// Clear the default transports

winston.clear()

winston.add(winston.transports.Console, {

level: 'debug',

colorize: true,

timestamp: true

})

// configure Logstash connection

winston.add(winston.transports.Logstash, {

level: 'debug',

port: 5000,

node\_name: 'exampleApp',

host: '34.67.20.170',

app\_name: 'XXX',

test\_message: 'aaaaaaaaaa'

});

module.exports = winston

* Import module và sử dụng như ví dụ sau

let logger = require('./logger')

logger.debug(‘Test message’);

* 1. **View log trên Kibana**
* Mở browser truy cập địa chỉ (kibana port : 5601):

[http://34.67.20.170:5601](http://34.67.20.170:5601/)

* Bắt đầu tra cứu log.