ELK---STACK

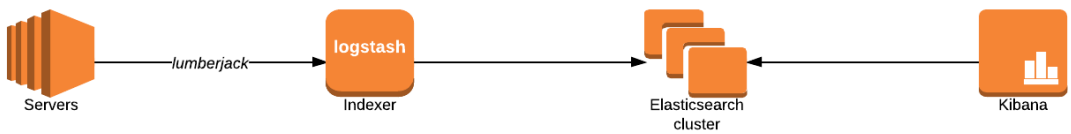
Cài đặt và sử dụng Logstash, elasticsearch, kibana để thu thập log

#Giới thiệu bộ công cụ ELK - Elastichsearch, Logstash, Kibana

- **Elastichsearch**: - RESTful distributed search engine, sử dụng NoSQL database dựa trên nền tảng Apache Lucene engine. Được phát triển bởi công ty Elastichsearch, công ty này cũng là chủ sở hữu của Kibana và Logstash
- **Logstash**: là một công cụ sử dụng để thu thập, xử lý log, được viết bằng Java, nhiệm vụ chính của logstash là thu thập log, phân tích nó (đánh index) sau đó lưu trữ tất cả vào trong Elasticsearch
- **Kibana**: là web interface sử dụng để tìm kiếm, hiển thị dữ liệu lưu trữ trong Elastichsearch.
- Logstash-forwarder (tên gọi trước đây là Lumberjack) là một trong rất nhiều "shipper" dùng để đẩy log đến server log tập trung, có nhiều tính năng đặc biệt sau
  - Nhẹ và dễ sử dụng (viết bằng Go, không cần JVM)
  - Sử dụng mật mã để truyền dữ liệu trên đường truyền

#KIẾN TRÚC CÁC THÀNH PHẦN

Đây là biểu đồ mô tả kiến trúc hoạt động của các thành phần



Ngoài các thành phần đã phân tích ở trên, còn một thành phần khá quan trọng nữa là broker, broker là thành phần ở giữa client với server, broker trong mô hình ELK thường sử dụng nhất là Redis, Redis có tác dụng giữ tất cả các bản tin log được đưa đến từ phía client, đưa nó vào hàng đợi, sau đó mới chueyern tiếp đến logstash. Điều này tránh được mất mát cho các bản tin log khi diễn ra tình trạng thắt nút cổ chai trong quá trình truyền tin.

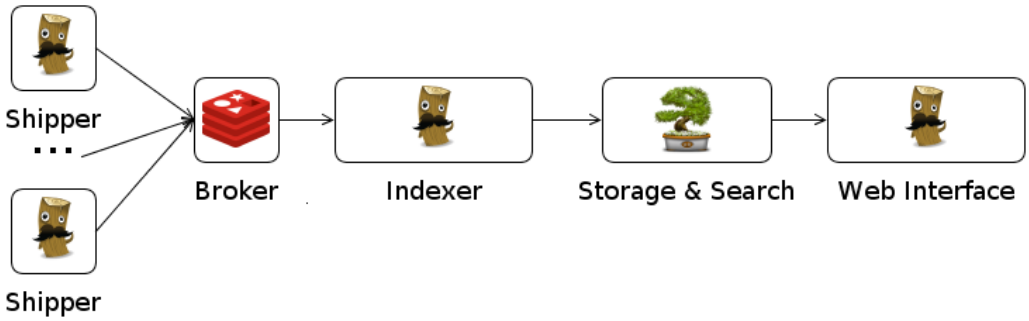
Một quá trình thu thập log được diễn ra như sau:

1. Các shipper (có nhiều loại shipper như: rsyslog, logstash-forwarder...) từ client đẩy log đến Broker , Broker xử lý log cho nó vào một hàng đợi trước khi nó được chuyển tiếp đến Logstash.

2.Logstash xử lý các bản tin log đến và lưu trữ nó dưới dạng JSON document vào trong Elastichsearch

3. Elastichsearch có nhiệm vụ chính là lưu trữ và tìm kiếm tất cả các dữ liệu

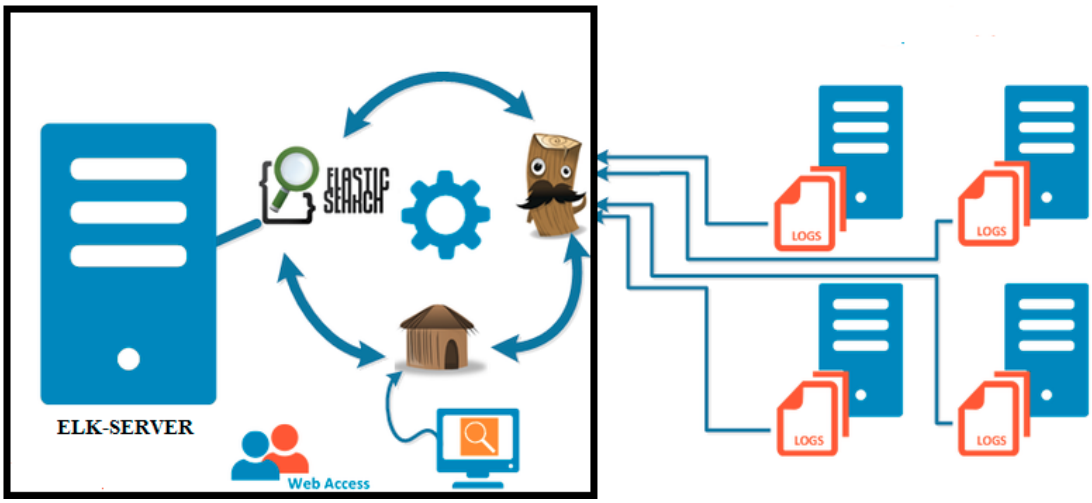
4. Sau đó bản tin log được hiển thị trên Kibana web interface



#Cài đặt:

Mô hình và các thành phần cài đặt

- Logstash: Logstash 1.4.1
- Elashticsearch: 1.4.4
- Kibana 3
- Logstash fowarder: 0.4.0



##ELK-Server

####1. Cài đặt Java 8

Do Logstash được viết bằng Java nên cần phải có một JVM để nó có thể hoạt động.

```
sudo add-apt-repository -y ppa:webupd8team/java
sudo apt-get update
sudo apt-get -y install oracle-java8-installer
```

####2. Cài đặt Elastichsearch

Elastichsearch là công cụ để lưu trữ, tìm kiếm mạnh mẽ. Là phần mềm quan trọng nhất trong bộ 3 sản phẩm ELK stack

```
wget -O - http://packages.elasticsearch.org/GPG-KEY-elasticsearch | sudo apt-key add -
echo 'deb http://packages.elasticsearch.org/elasticsearch/1.4/debian stable main' | sudo tee /etc/apt/so
sudo apt-get update
sudo apt-get -y install elasticsearch=1.4.4
sudo service elasticsearch restart
sudo update-rc.d elasticsearch defaults 95 10
```

####3. Cài đặt Kibana 3

Kibana là một giao diện web đầu cuối, nhà sản xuất sẽ cung cấp cho người dùng mã nguồn, do đó cần có một web server để thực thi mã nguồn này, ở đây mình sử dụng apache2

```
sudo apt-get install apache2 -y
sudo service apache2 restart
sudo service apache2 restart
```

Sau đó Download và config kibana

```
apt-get install unzip -y
cd ~; wget http://download.elasticsearch.org/kibana/kibana/kibana-latest.zip
unzip kibana-latest.zip
sudo mkdir -p /var/www/kibana
sudo cp -R ~/kibana-latest/* /var/www/kibana/
```

Cấu hình lại file `/etc/apache2/conf-enabled/kibana.conf` với nội dung sau

```
Alias /kibana /var/www/kibana
<Directory /var/www/kibana>
  Order allow,deny
  Allow from all
</Directory>
```

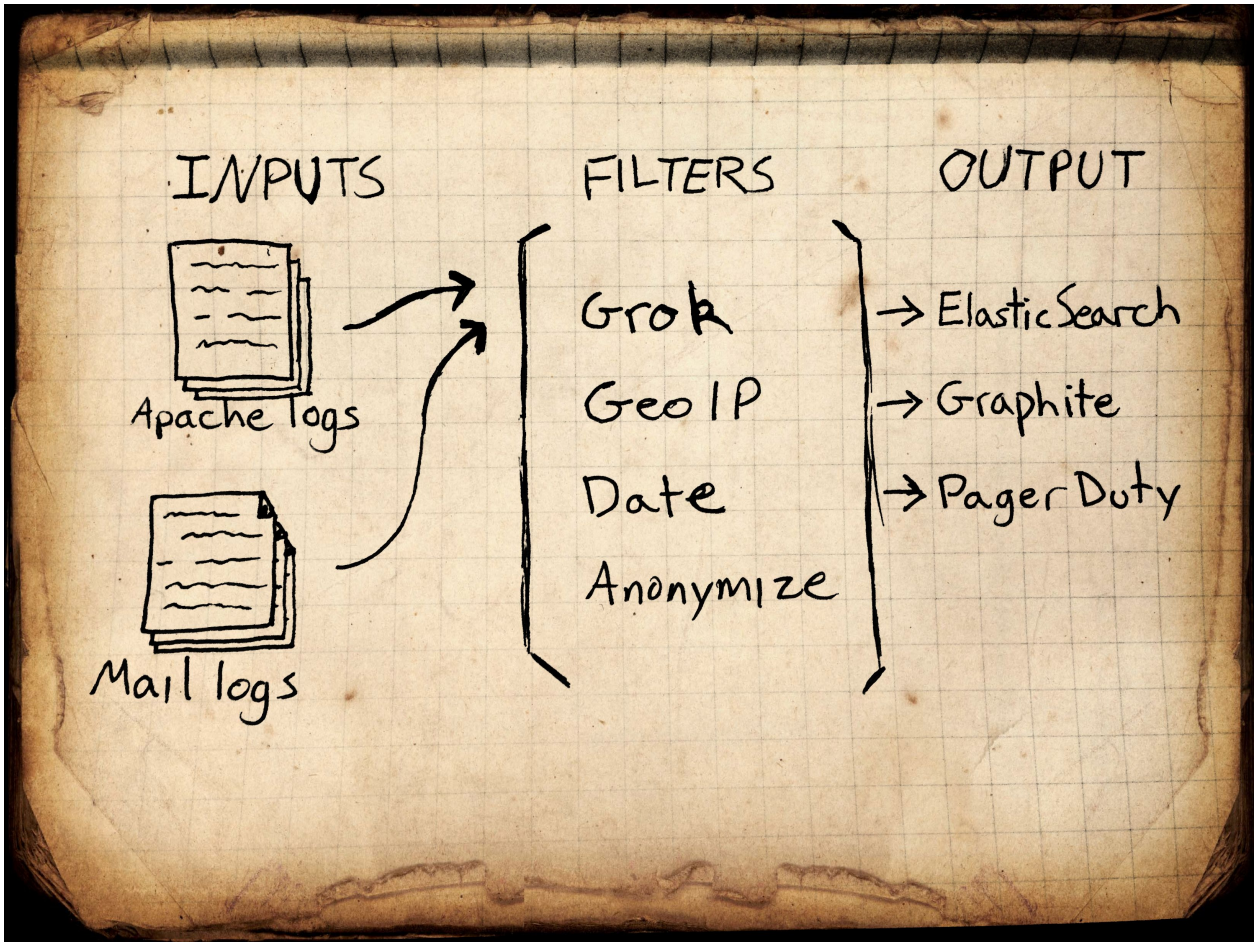
```
sudo service apache2 restart
```

####4. Cài đặt và cấu hình Logstash

```
echo 'deb http://packages.elasticsearch.org/logstash/1.4/debian stable main' | sudo tee /etc/apt/sources
sudo apt-get update
sudo apt-get install -y logstash=1.4.1-1-bd507eb --force-yes
```

Trước khi bắt đầu cấu hình Logstash, cần phải biết về cách nó xử lý một bản tin nhận đến thế nào

Có 3 thành phần chính trong file config của logstash là: **`INPUT, FILTER, OUTPUT`**



- **INPUT** có thể đến từ nhiều nguồn (từ rsyslog, logstash forwarder, logstash-agent, Redis broker, file, collectd,...) => Sau đó được **FILTER** - lọc (bằng cách thêm các trường vào bản tin nhận được, hay phân tích lại bản tin để thành các trường riêng biệt tùy ý...) => **OUTPUT** ra Elastichsearch để lưu trữ.

Có thể cấu hình 3 thành phần chính này trong cùng 1 file cấu hình duy nhất. Nhưng để dễ dàng quản lý mình sẽ cấu hình thành các file riêng biệt.

Chú ý Ở bài viết này tại phía client mình sử dụng **Logstash fowarder**, do mình đã nói ở trên, **Logstash fowarder sử dụng mật mã để bảo vệ dữ liệu truyền đi** nên trước cần phải tạo ra một cặp khóa SSL để xác thực giữa 2 bên gửi và nhận.

Tại **ELK server**

- Cấu hình file openssl.cnf

```
sudo vi /etc/ssl/openssl.cnf
```

- Tại [v3_ca] section

```
subjectAltName = IP: IP_của_ELK_Server
```

```
sudo mkdir -p /etc/pki/tls/certs
sudo mkdir /etc/pki/tls/private
cd /etc/pki/tls
sudo openssl req -config /etc/ssl/openssl.cnf -x509 -days 3650 -batch -nodes -newkey rsa:2048 -keyout pr
```

- Ở bước thực hiện trên, ở server sẽ tạo ra một certificate, bước tiếp theo là cần phải copy certificate này sang phía client:

```
scp /etc/pki/tls/certs/logstash-forwarder.crt user@client_server_private_address:/tmp
```

- Sau khi hoàn tất các bước cần thiết, bắt đầu cấu hình logstash:

- Tạo **INPUT** file

```
vi /etc/logstash/conf.d/02-logstashforwarder.conf
```

```
input {
  lumberjack {
    port => 5000
    type => "logs"
    ssl_certificate => "/etc/pki/tls/certs/logstash-forwarder.crt"
    ssl_key => "/etc/pki/tls/private/logstash-forwarder.key"
  }
}
```

- Tạo **FILTER** file

```
vi /etc/logstash/conf.d/20-filter.conf
```

```
filter {
  if [type] == "syslog" {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp} %{SYSLOGHOST:syslog_hostname} %{DATA:
      add_field => [ "received_at", "%{@timestamp}" ]
      add_field => [ "received_from", "%{host}" ]
    }
    syslog_pri { }
    date {
      match => [ "syslog_timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss" ]
    }
  }
}
```

- Tạo **OUTPUT** file

```
vi /etc/logstash/conf.d/30-output.conf
```

```
output {
  elasticsearch {
                host => localhost }
  stdout { codec => rubydebug }
}
```

##Tại Client

Cài đặt logstash fowarder và copy cert CA vào thư mục riêng

```
echo 'deb http://packages.elasticsearch.org/logstashforwarder/debian stable main' | sudo tee /etc/apt/so
wget -O - http://packages.elasticsearch.org/GPG-KEY-elasticsearch | sudo apt-key add -
sudo apt-get update
sudo apt-get install logstash-forwarder
sudo mkdir -p /etc/pki/tls/certs
sudo cp /tmp/logstash-forwarder.crt /etc/pki/tls/certs/
```

Cấu hình logstash fowarder

```
sudo vi /etc/logstash-forwarder.conf
```

```
{
  "network": {
        "servers": [ "IP_ELKSERVER:5000" ],
        "timeout": 15,
        "ssl ca": "/etc/pki/tls/certs/logstash-forwarder.crt"
  },

  "files": [
     {
      "paths": [
        "/var/log/syslog",
        "/var/log/auth.log"
       ],
      "fields": { "type": "syslog" }
```

```
      }

  #,
  #    {
  #
  #      "paths": [
  #       "/var/log/auth.log"
  #       ],
  #      "fields": { "type": "ssh" }
  #    }
  #,
  #    {
  #      "paths": [
  #        "/var/log/apache2/access.log"
  #       ],
  #      "fields": { "type": "apache-access" }
  #    }
  ]
  }

sudo service logstash-forwarder restart
```

Truy cập vào địa chỉ `http://ip-elk-server/kibana