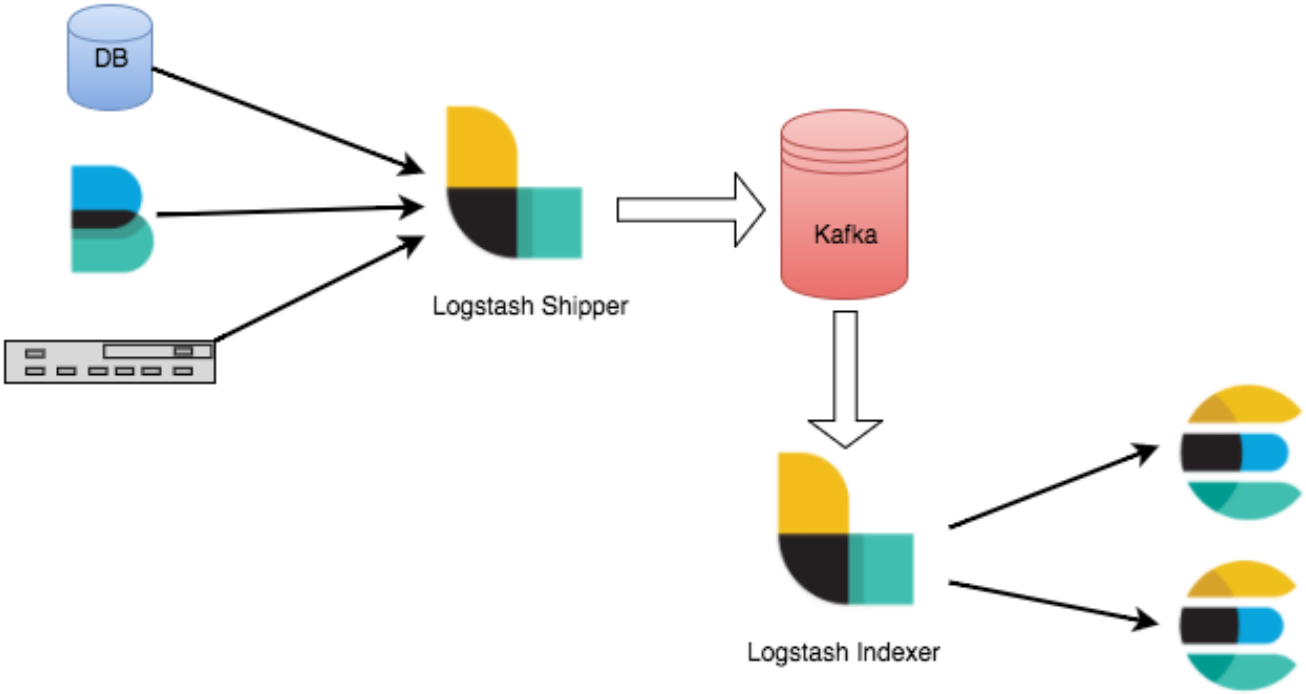
Shipper/Indexer



Shipper

- 다른 데이터 소스로부터 데이터를 수신
- 데이터 처리를 수행하지 않음

> shipper
> Shipper의 책임은 Kafka 토픽, 즉 생산자로 수신되는 데이터를 즉시 전송하는 것

Indexer

- 데이터를 소비
- Grok, DNS 조회, Elasticsearch 인덱싱과 같은 대량 변환을 실행

> indexer
> Indexer는 Kafka 토픽을 전처리 수행 후 Elasticsearch로 전송하는 것

설치

- Elasticsearch
- Kibana
- Logstash
- Filebeat
- Kafka

설정

1. FileBeats 설정

filebeats 설정 초기화

```
> filebeat -e setup
```

filebeat.yml output 수정

```
#output.elasticsearch:
  #hosts: ["localhost:9200"]


output.logstash:
  hosts: ["localhost:5044"]
```

module apache 설정

```
> filebeat modules enable apache
```

modules.d/apache.yml 수정

```
#access_log
var.paths: ["/private/var/log/apache2/access_log*"]

#error_log
var.paths: ["/private/var/log/apache2/error_log*"]
```

2. Logstash(Shipper)

config/logstash-shipper.conf 생성

```
input {
  beats {
    port => 5044
  }
}

output {
  kafka {
    bootstrap_servers => "127.0.0.1:9092"
    topic_id => "apache-log"
    codec => json
  }
  stdout {
    codec => json
  }
}
```

3. Logstash(Indexer)

config/logstash-indexer.conf 생성

```
input {
  kafka {
    bootstrap_servers => "127.0.0.1:9092"
    topics => "apache-log"
    group_id => "apache-log-logstash"
    codec => json
  }
}

filter {
  if [fileset][module] == "apache2" {
    if [fileset][name] == "access" {
      grok {
        match => { "message" =>
["%{IPORHOST:[apache2][access][remote_ip]} -
%{DATA:[apache2][access][user_name]}
\[%{HTTPDATE:[apache2][access][time]}\]
\"%{WORD:[apache2][access][method]} %{DATA:[apache2][access][url]}
HTTP/%{NUMBER:[apache2][access][http_version]}\"
%{NUMBER:[apache2][access][response_code]}
%{NUMBER:[apache2][access][body_sent][bytes]}(
\"%{DATA:[apache2][access][referrer]}\")?(
\"%{DATA:[apache2][access][agent]}\")?",
          "%{IPORHOST:[apache2][access][remote_ip]} -
%{DATA:[apache2][access][user_name]}
\\[%{HTTPDATE:[apache2][access][time]}\\] \"-\"
%{NUMBER:[apache2][access][response_code]} -" ] }
        remove_field => "message"
      }
      mutate {
        add_field => { "read_timestamp" => "%{@timestamp}" }
      }
      date {
        match => [ "[apache2][access][time]", "dd/MMM/YYYY:H:m:s Z"
]
        remove_field => "[apache2][access][time]"
      }
      useragent {
        source => "[apache2][access][agent]"
        target => "[apache2][access][user_agent]"
        remove_field => "[apache2][access][agent]"
      }
      geoip {
        source => "[apache2][access][remote_ip]"
        target => "[apache2][access][geoip]"
      }
    }
    else if [fileset][name] == "error" {
      grok {
        match => { "message" =>
["\[%{APACHE_TIME:[apache2][error][timestamp]}\]
\[%{LOGLEVEL:[apache2][error][level]}\]( \[client
%{IPORHOST:[apache2][error][client]}\])?
%{GREEDYDATA:[apache2][error][message]}",
```

```
          "\[%{APACHE_TIME:[apache2][error][timestamp]}\]
\[%{DATA:[apache2][error][module]}:%{LOGLEVEL:[apache2][error][leve
l]}\] \[pid %{NUMBER:[apache2][error][pid]}(:tid
%{NUMBER:[apache2][error][tid]})?\]( \[client
%{IPORHOST:[apache2][error][client]}\])?
%{GREEDYDATA:[apache2][error][message1]}" ] }
        pattern_definitions => {
          "APACHE_TIME" => "%{DAY} %{MONTH} %{MONTHDAY} %{TIME}
%{YEAR}"
        }
        remove_field => "message"
      }
      mutate {
        rename => { "[apache2][error][message1]" =>
"[apache2][error][message]" }
      }
      date {
        match => [ "[apache2][error][timestamp]", "EEE MMM dd H:m:s
YYYY", "EEE MMM dd H:m:s.SSSSSS YYYY" ]
        remove_field => "[apache2][error][timestamp]"
      }
    }
  }
}

output {
  elasticsearch {
    hosts => localhost
    manage_template => false
    index =>
"%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
  }
  stdout {
```

```
    codec => json
  }
}
```

실행

1. Elasticsearch 실행

Elasticsearch 실행 명령어

```
> bin/elasticsearch
```

2. Kibana 실행

Kibans 실행 명령어

```
> bin/kibana
```

3. Kafka 실행

Kafka 실행 명령어

```
> bin/zookeeper-server-start.sh config/zookeeper.properties
> bin/kafka-server-start.sh config/server.properties


> bin/kafka-topics.sh --create --zookeeper localhost:2181
--replication-factor 1 --partitions 1 --topic apache-log


> bin/kafka-console-consumer.sh --bootstrap-server localhost:9092
--topic apache-log --from-beginning
```

LEADER_NOT_AVAILABLE 오류 발생
`config/server.properties` advertised.listeners = PLAINTEXT://localhost:9092

4. Indexer 실행

Indexer 실행 명령어

```
> bin/logstash -f ./config/logstash-indexer.conf
```

5. Shipper 실행

Shipper 실행 명령어

```
> bin/logstash -f ./config/logstash-shipper.conf
```

6. FileBeats 실행

FileBeats 실행 명령어

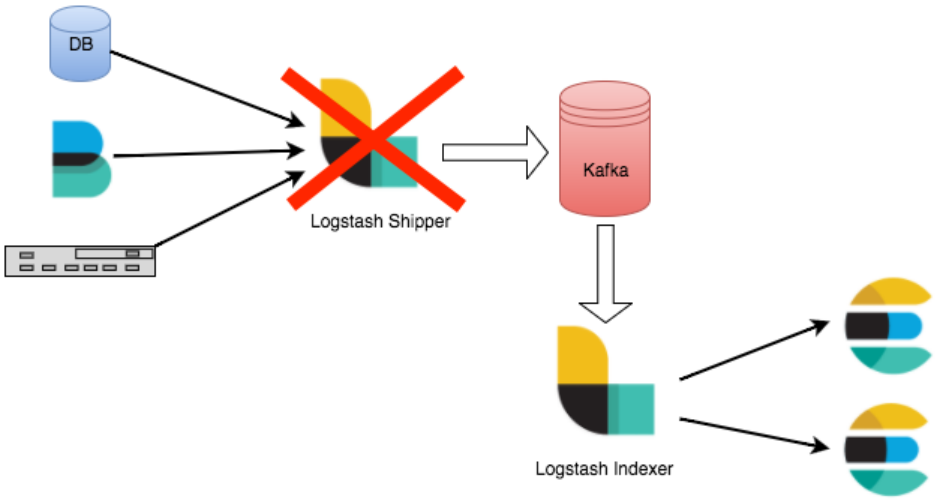
```
> filebeat -e
```

확인

http://127.0.0.1:5601

부록

1. FileBeats → Kafka



filebeat.yml 수정

```
output.kafka:
  hosts: ["localhost:9092"]
  topic: "apache-log"
  codec.json:
    pretty: false
```

2. Ingest Node pipeline 사용하기

filebeat setup pipeline

```
> filebeat setup --pipelines --modules apache
```

위 명령어 실행시에는 filebeat.yml의 output이 Elasticsearch 이어야 함

logstash-pipeline-indexer.config 생성

```
input {
  kafka {
    bootstrap_servers => "localhost:9092"
    topics => ["apache-log"]
    codec => json
  }
}

output {
  if [@metadata][pipeline] {
    elasticsearch {
      hosts => localhost
      manage_template => false
      index =>
"%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
      pipeline => "%{[@metadata][pipeline]}"
    }

    stdout {
      codec => json
    }
  } else {
    elasticsearch {
      hosts => localhost
      manage_template => false
      index =>
"%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
    }

    stdout {
      codec => json
    }
  }
}
```