05-logstash-config.yml 이해하기

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: logstash-config
  namespace: logging
```

- kind: ConfigMap
- metadata: logging 네임스페이스에 logstash-config ConfigMap 생성합니다

```
data: # 실제 설정 내용
  logstash.conf: | # logstash.conf 키 아래에 파이프라인 설정 내용 저장
    input { # Logstash가 데이터 받아오는 위치 정의 섹션
      beats {
        # 5044 포트 열고 Filebeat가 데이터 보내주기 기다림
        # Filebeat의 output.logstash 설정 포트와 일치해야 한다.
        port => 5044
      }
    }
    filter {
      # 1. Spring Boot 로그인지 확인 (컨테이너 이름에 'service'가 포함된 경우)
      if "service" in [kubernetes][container][name] {
        grok {
          match => { "message" => "%{TIMESTAMP_ISO8601:log_timestamp}\s+%
{LOGLEVEL:log_level}\s+%{NUMBER:pid}\s+---\s+\[%{DATA:service_name}\]\s+\[\s*%
{DATA:thread_name}\]\s+%{JAVACLASS:class_name}\s+:\s+%{GREEDYDATA:log_message}" }
          # 파싱 성공 시 태그 추가
          add_tag => [ "grok_parsed_spring" ]
        }
        date {
          match => [ "log_timestamp", "ISO8601" ]
          target => "@timestamp"
        }
      }
      # 2. 다른 종류의 로그(예: Nginx)를 위한 파싱 규칙을 여기에 추가할 수 있습니다.
      # else if [kubernetes][container][name] == "nginx" {
      #   grok {
      #     match => { "message" => "%{IPORHOST:clientip} %{USER:ident} %
{USER:auth} \[%{HTTPDATE:timestamp}\] \"%{WORD:verb} %{URIPATHPARAM:request}
HTTP/%{NUMBER:httpversion}\" %{NUMBER:response:int} %{NUMBER:bytes:int}" }
      #     add_tag => [ "grok_parsed_nginx" ]
      #   }
      # }

      # 3. 위 조건에 맞지 않는 로그는 JSON 형식으로 파싱 시도 (범용적인 처리)
      else {
        json {
```

```
            source => "message"
            # JSON 파싱 성공 시 태그 추가
            add_tag => [ "json_parsed" ]
          }
        }

        # 파싱에 성공했다면(grok 또는 json), 원래의 긴 message 필드는 삭제
        if "grok_parsed_spring" in [tags] or "grok_parsed_nginx" in [tags] or
"json_parsed" in [tags] {
          mutate {
            remove_field => ["message"]
          }
        }
      }
      output {
        elasticsearch {
          hosts => ["http://host.docker.internal:9200"]
          index => "project-1-logs-%%{+YYYY.MM.dd}"
        }
      }
```

> 필터 기능을 사용하지 않는 경우, 로그가 단순한 긴 문자열로 표시됩니다.

- "로그 레벨이 ERROR인 로그만 찾기", "특정 클래스에서 발생한 로그만 조회하기" 등의 검색 기능을 사용하기 위해서 위와 같이 Logstash의 grok 필터를 사용해 볼 수 있습니다.
- 현재는 많이 사용하는 Spring boot 기반 서비스 로그에 대해 필터를 정의했는데, 주석처리된 내용을 참고해서 다른 플랫폼을 지원하는 필터도 추가 적용할 수 있습니다.
- default namespace에 Elasticsearch가 배포돼 있다면 hosts로 다음 주소를 사용하면 됩니다.
    - "http://elasticsearch.default.svc.cluster.local:9200"

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: logstash-yml-config
  namespace: logging
data:
  logstash.yml: |
    http.host: "0.0.0.0"
    xpack.monitoring.enabled: false
```

- Logstash의 기본 모니터링 기능을 끄는 설정을 담고 있습니다.
- Elasticsearch 주소 불일치와 관련한 불필요한 경고 로그를 제거하기 위해 사용합니다.
    - Logstash 부가 기능인 내부 모니터링에서 기본 설정값인 elasticsearch:9200 을 찾지 못하면 에러 메시지가 남습니다.