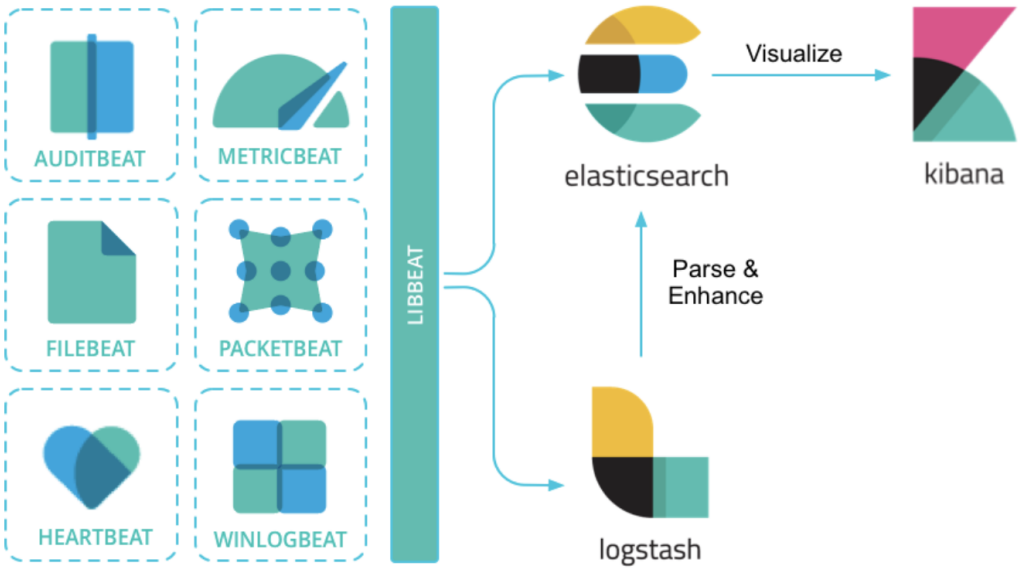
[6주차] 7. 비츠

> 비츠는 수집한 데이터를 엘라스틱으로 실어 나르기 위해 최전방에 위치한 소프트웨어다.

- 가볍고 사용하기 쉬운 데이터 수집기
- Go언어로 작성된 경량 프로그램
- 오픈소스 라이선스로 Libbeat 라는 프레임워크를 제공한다.

1. 비츠 소개

- 로그스태시는 다양한 플러그인을 포함하므로 범용성이 높지만 무겁게 움직인다
- 비츠는 범용성은 낮지만 특정 목적만 수행하도록 가벼워 애플리케이션 성능에 영향을 미치지 않고 필요한 이벤트를 수집할 수 있다.
- 비츠에서 수집한 데이터를 ES 로 바로 보내거나 로그스태시를 거쳐 ES로 보낸다.



- 비츠는 로그스태시의 대체재라기보다 공생관계라고 할 수 있다.

▼ 비츠 종류

- 파일비트 : 로그 파일을 실시간으로 읽어서 전송한다. 로그스태시나 엘라스틱서치에 데이터 전달 시 부하 방지 기능을 지원한다. 가장 많이 사용되는 비츠
- 하트비트: 서비스가 살아 있는지 확인하고 모니터링한다. 핑, HTTP 서비스 체크 등 서비스 상태를 파악하기 위한 여러 방법들을 지원하며 크론 스케줄 표현식을 기반으로 주기적으로 서비스 활성여부를 판단하다.
- 메트릭비트: 운영체제를 포함해 서비스로부터 주기적으로 통계 지표를 수집한다. 시스템의 CPU나 메모리같은 정보, 데이터베이스, 프록시 같은 특정 서비스의 통계 데이터도 수집할 수 있다.
- 패킷비트: 네트워크 패킷 분석기로 네트워크 데이터를 수집한다. 트래픽, 응답 시간, 사용자 패턴 분석 같은 작업 등을 할 수 있다.
- 윈로그비트: 윈도우에서 발생하는 이벤트 로그를 수집한다.
- 오딧비트: 리눅스의 시스템 보안 정보를 감사하는 auditd 에서 생성하는 로그를 수집한다. 사용자가 사전에 정의한 규칙에 따르지 않는 로그들을 식별해 계정 미준수나 보안 정책에 위배되는 동작을 식별할 수 있다.
- 저널비트: 리눅스 systemd로 실행되는 서비스들에 대한 로그를 수집한다.
- 펑션비트: 클라우드의 서버리스 환경에서 사용할 수 있고 아마존의 큐 데이터 등을 수집한다.
- 카스텀 비츠: 오픈소스 커뮤니티에서 만든 비츠.
  - 카프카비트: 카프카 토픽 데이터 수집
  - 엔진엑스비트: 엔진엑스 상태 수집
  - MySQ비트: MySQL 상태 수집

2. 비츠 설치

https://www.elastic.co/kr/downloads/beats 에서 다운로드할 수 있다.

- 비트 동작 과정
  1. 비트 다운
  2. 비트 설정 파일 수정

3. 엘라스틱서치와 키바나 대시보드를 사용할 수 있게 설정
4. 비트 시작
5. 키바나 대시보드에서 데이터 확인

3. 파일비트

1. 파일비트 아키텍쳐
   - 입력: 설정 파일에서 하베스터에 대한 **입력 소스**를 정한다. 파일비트는 하나 혹은 여러 개의 입력을 가질 수 있다.
   - 하베스터(harvester): 입력에 명시된 파일을 직접 **수집하는 주체**다. 파일은 하나의 하베스터를 가지며, 하베스터는 파일을 한 줄 씩 읽고 내보내는 역할을 한다. 파일을 열고 닫는 역할도 한다. 하베스터가 실행되는 동안에는 파일 디스크립터가 열려 있다.
   - 스풀러(spooler): 하베스터가 수집한 이벤트를 엘라스틱서치나 로그스태시 같은 장소로 **전달**한다.

   ⇒ 입력에서 대상 경로를 모니터링하다가 새로운 파일이 발견되면 하베스터를 생성해 해당 데이터를 읽는다.
2. 파일비트 실행

```
filebeat.inputs:
- type: log
  enabled: false
  paths:
    - /var/log/*.log
output.elasticsearch:
  hosts: ["localhost:9200"]
setup.kibana:
  host: "localhost:5601"
```

   - 파일비트 인풋 타입
     - log: 파일 시스템의 지정한 경로에서 로그 파일을 읽어 들인다
     - container: 도커같은 컨테이너의 로그를 수집하기 위한 입력으로 파일을 읽어 들인다는 점에서 log와 유사하다.
     - s3: log 타입과 유사하나 아마존 웹 서비스의 S3버킷에 위치한 파일을 읽어 들인다.

    - kafka: 카프아의 토픽을 읽어들인다.
- 파일비트 아웃풋 타입
    - elasticsearch: 가장많이 사용되는 타입, 수집한 이벤트를 엘라스틱서치로 직접 인덱싱한다.
    - logstash: 다수의 비츠를 사용해 엘라스틱서치로 전송되는 인덱싱 리퀘스트의 양이 많거나, 비츠나 인제스트 노드 수준에서 처리하기 어려운 가공 작업이 필요할 때 별도의 로그 스태시를 구축한 후 수집한 이벤트를 전송한다. 다수의 인덱싱 요청이 로그 스태시에서 단일 벌크 리퀘스트로 묶여 인덱싱 효율의 개선을 기대할 수 있다. 이때 전송한 이벤트는 로그스태시의 beats인풋을 이용해 입력 받을 수 있다.
    - kafka: 파일비트에서 1차적으로 수집한 이벤트를 카프카로 전송한다. 카프카는 좀 더 안정적인 수집 파이프라인을 구성할 때 신뢰할 만한 중간 저장소/큐이므로 수집 중 장애 발생 시 데이터 손실을 최소화 하기 위한 방안으로 활용된다. 최종적으로 엘라스틱서치의 인덱싱을 원할 경우 이후 다시 파일비트의 카프카 인풋을 이용하거나 로그스태시의 카프카 인풋을 이용해 입력할 수 있다.
    - console: 수집한 이벤트를 시스템 콘솔에 출력한다. 일반적으로 수집이 정상적으로 이뤄지는지 입력 설정을 테스트하기 위한 목적으로 사용된다.

3. 파일 비트 설정
    - 유용한 설정
        - ignore_older: 새로운 파일 탐색 시 오래된 파일은 읽어들이지 않고 무시한다, 타임스트링 형식으로 작성
        - include_lines: 특정 라인을 정규 표현식을 이용해 필터링하고 매칭된 라인만 비츠에서 수용 한다.
        - exclude_lines: 특정 라인을 정규 표현식을 이용해 필터링하고 매칭된 라인은 내부에서 버린다.
        - exclude_files: 정규표현식 패턴에 일치하는 파일을 무시한다.
    - 멀티라인 로그 처리
        - pttern: 패턴을 지정하고 일치하는 라인이면 멀티라인으로 인식한다.
        - negate: true 일 때 패턴 일치 조건을 반전시킨다.
        - match: before/after를 지정해서 멀티라인을 처리하는 방식이다.

4. 모듈

- 많이 사용되고 잘 알려진 시스템 데이터를 수집하기 위한 일반적인 설정을 사전 정의해 둔 것
- 비트 파일 설정을 수정후 모듈을 활성화하고 모듈 설정 파일을 수정한다.

```
filebeat.config.modules:
  pathL ${path.config}/modules.d/*.yml
```

```
$ ./filebeat modules enable logstash #모듈활성화
$ ./filebeat modules enable logstash kibana elasticsearch #멀티 모듈활성화
$ ./filebeat modules disable logstash kibana elasticsearch #멀티 모듈 비활성화
$ ./filebeat modules list #활성화된 모듈 확인
```

- aws: AWS의 CloudWatch, CloudTrail 같은 서비스에서 발생하는 로그들을 수집할 수 있다.
- cef: 시스로그를 통해 CEF(Common Event Format) 이벤트를 입력받을 수 있다.
- cisco: 시스코사의 ASA, Nexus 등 네트워크 장비에서 발생되는 이벤트를 수집한다.
- elasticsearch: 엘라스틱서치의 클러스터 GC 감사로그 등을 수집할 때 사용된다.
- googlecloud: 구글 클라우드 플랫폼의 VPC 플로우, 방화벽 로그 등을 수집할 수 있다.
- logstash: 로그 스태시에서 발생한 로그들을 수집할 수 있다.

4. 모니터링

- 비츠가 데이터 입출력을 잘하고 있는지, 호스트의 리소스들은 문제 없는지 등을 엘라스틱서치와 키바나를 통해 확인할 수 있는 기능

```
monitoring.enabled:true
# monitoring.cluster_uuid: PRODUCTION_ES_CLUSTER_UUID
monitoring.elasticsearch:
  hosts: ["localshot:9220"]
```